AIED
2013
MEMPHIS

AIED 2013: 16th International Conference on Artificial Intelligence in Education

Workshops Proceedings

Edited by:

Erin Walker
Arizona State University, USA

Chee-Kit Looi
Nanyang Technological University, Singapore

July 9-13,
Memphis, Tennessee, USA

Preface

The supplementary proceedings of the workshops held in conjunction with AIED 2013, the sixteen International Conference on Artificial Intelligence in Education, July 9-13, 2013, Memphis, Tennessee, USA are organized as a set of volumes - a separate one for each workshop.

The set contains the proceedings of the following workshops:

- **Volume 1: Workshop on Massive Open Online Courses (moocshop)**
  Co-chairs: Zachary Pardos & Emily Schneider
  http://www.moocshop.org
- **Volume 2: Scaffolding in Open-Ended Learning Environments (OELEs)**
  Co-chairs: Gautam Biswas, Roger Azevedo, Valerie Shute, & Susan Bull
  https://sites.google.com/site/scaffoldingoeles/home
- **Volume 3: 2nd Workshop on Intelligent Support for Learning in Groups**
  Co-chairs: Jihie Kim & Rohit Kumar
  https://sites.google.com/site/islg2013/
- **Volume 4: AIED Workshop on Simulated Learners**
  Co-chairs: Gord McCalla & John Champaign
  https://sites.google.com/site/aiedwsl
- **Volume 5: 4th International Workshop on Culturally-Aware Tutoring Systems (CATS2013)**
  Co-chairs: Emmanuel G. Blanchard & Isabela Gasparini
  http://cats-ws.org/
- **Volume 6: CrossCultural Differences and Learning Technologies for the Developing World (LT4D) – Issues, Constraints, Solutions**
  Co-chairs: Ivon Arroyo, Imran Zualkernan, & Beverly P. Woolf
  http://cadmium.cs.umass.edu/LT4D/Welcome.html
- **Volume 7: Recommendations for Authoring, Instructional Strategies and Analysis for Intelligent Tutoring Systems (ITS): Toward the Development of a Generalized Intelligent Framework for Tutoring (GIFT)**
  Co-chairs: Robert A. Sottilare & Heather K. Holden
  https://gifttutoring.org/news/14
- **Volume 8: Formative Feedback in Interactive Learning Environments (FFILE)**
  Co-chairs: Ilya Goldin, Taylor Martin, Ryan Baker, Vincent Aleven, & Tiffany Barnes
  http://sites.google.com/site/ffileworkshop/

- **Volume 9: The First Workshop on AI-supported Education for Computer Science (AIEDCS)**
  Co-chairs: Barbara Di Eugenio, Sharon I-Han Hsiao, Kristy Elizabeth Boyer, Nguyen-Thinh Le, & Leigh Ann Sudol-DeLyser
  https://sites.google.com/site/aiedcs2013/
- **Volume 10: Workshop on Self-Regulated Learning in Educational Technologies (SRL@ET): Supporting, modeling, evaluating, and fostering metacognition with computer-based learning environments**
  Co-chairs: Amali Weerasinghe, Benedict Du Boulay, & Gautam Biswas
  http://workshops.shareghi.com/AIED2013/

While the main conference program presents an overview of the latest mature work in the field, the AIED2013 workshops are designed to provide an opportunity for in-depth discussion of current and emerging topics of interest to the AIED community. The workshops are intended to provide an informal interactive setting for participants to address current technical and research issues related to the area of Artificial Intelligence in Education and to present, discuss, and explore their new ideas and work in progress.

All workshop papers have been reviewed by committees of leading international researchers. We would like to thank each of the workshop organizers, including the program committees and additional reviewers for their efforts in the preparation and organization of the workshops.

May, 2013
Erin Walker & Chee-Kit Looi

AIED 2013 Workshops Proceedings
Volume 1

Workshop on Massive Open Online Courses (moocshop)

Workshop Co-Chairs:

Zachary A. Pardos
Massachusetts Institute of Technology

Emily Schneider
Stanford University

http://www.moocshop.org

Preface

The moocshop surveys the rapidly expanding ecosystem of Massive Open Online Courses (MOOCs). Since late 2011, when enrolment for Stanford's AI class went viral, MOOCs have been a compelling and controversial topic for university faculty and administrators, as well as the media and blogosphere. Research, however, has played a relatively small role in the dialogue about MOOCs thus far, for two reasons. The first is the quickly moving landscape, with course scale and scope as the primary drivers for many stakeholders. The second is that there has yet to develop a centralized space where researchers, technologists, and course designers can share their findings or come to consensus on approaches for making sense of these emergent virtual learning environments.

Enter the moocshop. Designed to foster cross-institutional and cross-platform dialogue, the moocshop aims to develop a shared foundation for an interdisciplinary field of inquiry moving forward. Towards this end, we invited researchers, technologists, and course designers from universities and industry to share their work on key topics, from analytics to pedagogy to privacy. Since the forms and functions of MOOCs are continuing to evolve, the moocshop welcomed submissions on a variety of modalities of open online learning. Among the accepted papers and abstract-only submissions, four broad categories emerged:

- Position papers that proposed lenses for analyses or data infrastructure required to lower the barriers for research on MOOCs
- Exploratory analyses towards designing tools to assess and provide feedback on learner knowledge and performance
- Exploratory analyses and case studies characterizing learner engagement with MOOCs
- Experiments intended to personalize the learner experience or affect the psychological state of the learner

These papers and abstracts are an initial foray into what will be an ongoing dialogue, including discussions at the workshop and a synthesis paper to follow based on these discussions and the proceedings. We are pleased to launch the moocshop at the joint workshop day for AIED and EDM in order to draw on the expertise of both communities and ground the workshop discussions in principles and lessons learned from the long community heritage in educational technology research. Future instantiations of the moocshop will solicit contributions from a variety of different conferences in order to reflect the broad, interdisciplinary nature of the MOOC space.

June, 2013
Zachary A. Pardos & Emily Schneider

Program Committee

Co-Chair: Zachary A. Pardos, *MIT, USA* (pardos@mit.edu)
Co-Chair: Emily Schneider, *Stanford, USA* (elfs@cs.stanford.edu)

Ryan Baker, *Columbia Teacher's College, USA*
Amy Collier, *Stanford University, USA*
Chuong Do, *Coursera, USA*
Neil Heffernan, *Worcester Polytechnic Institute, USA*
Jack Mostow, *Carnegie Mellon University, USA*
Una-May O'Reilly, *Massachusetts Institute of Technology, USA*
Zach Pardos, *Massachusetts Institute of Technology, USA*
David Pritchard, *Massachusetts Institute of Technology, USA*
Emily Schneider, *Stanford University, USA*
George Siemens, *Athabasca University, USA*
John Stamper, *Carnegie Mellon University, USA*
Kalyan Veeramachaneni - *Massachusetts Institute of Technology, USA*

Table of Contents

Position / Overview

Assessment

Interventions

Engagement

Two Models of Learning: Cognition Vs. Socialization

Shreeharsh Kelkar[1]

[1] Massachusetts Institute of Technology
United States
skelkar@mit.edu

Abstract. In this paper, I bring out the contrasts between two different approaches to student learning: that of computational learning scientists and socio-cultural anthropologists, and suggest some implications and directions for learning research in MOOCs. Computational learning scientists see learning as a matter of imbibing particular knowledge propositions, and therefore understand teaching as a way of configuring these knowledge propositions in a way that takes into account the learner's capacities. Cultural anthropologists see learning as a process of acculturation or socialization--the process of becoming a member of a community. They see school itself as a social institution and the process of learning at school as a special case of socialization into a certain kind of learning style (Lave 1988); being socialized into this learning style depends on the kinds of social and cultural resources that a student has access to.

Rather than see these approaches as either right or wrong, I see them as productive leading to particular kinds of research: thus, while a computational model of learning leads to research that looks at particular paths through the course material that accomplish the learning of a concept, an anthropological approach would look at student-student and student-teacher forum dialog to see how students use language, cultural resources and the affordances of the forum itself to make meaning. I argue that a socialization approach to learning might be more useful for humanities courses where assignments tend to be essays or dialogue. Finally, I bring up the old historical controversy in Artificial Intelligence: between the Physical Symbol Systems hypothesis and situated action. I argue that some of the computational approaches taken up by the proponents of situated action may be useful exemplars to implement a computational model of learning as socialization.

Keywords: cultural anthropology, learning models, socialization

welcome to the moocspace: a proposed theory and taxonomy for massive open online courses

Emily Schneider[1]

[1] Lytics Lab, Stanford University,
Stanford, CA
elfs@cs.stanford.edu

Abstract. This paper describes a theoretical framework and feature taxonomy for MOOCs, with the goal of developing a shared language for researchers and designers. The theoretical framework characterizes MOOC design goals in terms of stances towards knowledge, the learner, and assessment practices, taking as a starting point the affordances of the Web and digital learning environments. The taxonomy encompasses features, course structures, and audiences. It can be mapped onto the theoretical framework, used by researchers to identify similar courses for cross-course comparisons, and by instructional designers to guide design decisions in different dimensions. Both the theory and the taxonomy are intended in the spirit of proposal, to be refined based on feedback from MOOC researchers, designers, and technologists.

Keywords: taxonomy, knowledge organization, MOOCs, online learning theory

1 Introduction

If learning is the process of transforming external information into internal knowledge, the Internet offers us a universe of possibilities. In this context, MOOCs are simply a well-structured, expert-driven option for openly accessible learning opportunities. As of mid-2013, the boundaries of the moocspace[1] remain contested, with opinions (data-driven or no) generated daily in the blogosphere, the mainstream media, and an increasing number of academic publications. Meanwhile, decisions being made at a breakneck speed within academic institutions, governmental bodies, and private firms. What of the earlier forms of teaching and learning should we bring forward with us into networked, digital space, even as its interconnected and virtual

[1] Other types of open online learning opportunities that lend themselves to be named with similar wordplay include the DIYspace (e.g. Instructables, Ravelry, MAKE Magazine), the Q-and-Aspace (e.g. Quora, StackOverflow), the OERspace (indexed by such services as OERCommons and MERLOT), the coursespace (freely available course syllabi and instructional materials that are not officially declared or organized as OER), and the gamespace (where to even begin?). Then there is Wikipedia, the blogosphere and newsites, curated news pages (both crowdsourced, e.g. Slashdot, and personalized, e.g. Pinterest), and the great morass of affinity groups and individual, information-rich webpages.

nature allow us to develop new forms? How can an interdisciplinary, distributed group of researchers, course designers, administrators, technologists, and commentators make sense of our collective endeavor?

Towards a shared language for the *how* and *what* we are creating with MOOCs, I offer two frameworks. Firstly, for orientation towards the goals we have when we design MOOCs, I propose a theoretical framework that characterizes our assumptions about knowledge, the learner, and assessments. The framework takes as a starting point the affordances of the Web and digital learning environments, rather than those of brick-and-mortar learning environments.

Secondly, for grounding in the concrete, I offer a taxonomy of MOOC features, structures, and audiences, designed to capture the broad scope of MOOCs in terms of lifelong learning opportunities. Each element of the taxonomy can be mapped onto the theoretical framework to make explicit the epistemological stances of designers. The taxonomy can be used by researchers as a way of identifying similar courses for cross-course comparisons, and by instructional designers as a set of guideposts for potential design decisions in different dimensions. Finally, in the closing section of the paper, I provide an example of mapping the theory onto features from the taxonomy and introduce an application of the taxonomy as the organizing ontology for a digital repository of research on MOOCs, also referred to as the moocspace. Each framework is meant as a proposal to be iterated upon by the community.

2 A Proposed Theory *(Orientation)*

MOOC criticism and design decisions have largely been focused on comparisons with brick-and-mortar classrooms: how do we translate the familiar into these novel digital settings? Can classroom talk be replicated? What about the adjustments to teaching made by good instructors in response to the needs of the class? It is imperative to reflect on what we value in in-person learning environments and work to maintain the nature of these interactions. But to properly leverage the networked, digital environment to create optimal learning opportunities for MOOC participants, we also need to compare the virtual to the virtual and explore opportunities to embody the core principles of cyberspace in a structured learning environment.

Techno-utopian visions for the Web have three dominant themes: participatory culture, personalization, and collective intelligence. Participatory culture highlights the low cost of producing and sharing digital content, enabled by an increasing number of authoring, curatorial, and social networking tools [1]. In this account, personal expression, engagement, and a sense of community are available to any individual with interest and time—an ideal that MOOCs have begun to realize with well-facilitated discussion boards, and somewhat, with peer assessment. Some individual courses have also encouraged learners to post their own work in a portfolio style. But overall there are not many activities in this vein that have been formalized in the moocspace.

Participatory culture's elevation of the self is echoed in the personalized infrastructure of Web services from Google to Netflix, which increasingly seek to use recommendation engines to provide customized content to all users. The algorithmic principles of this largely profit-driven personalization are extendable to learning environments, though desired outcomes for learning are more complex than the metrics used for business analytics--hence the need for learning analytics to develop robust and theory-driven learner models for adaptive environments. Visions of personalized digital learning include options for learners to engage with the same content at their own pace, or to be treated to differentiated instruction based on their preferences and goals [2]. In MOOCs this will require robust learner models based on interaction data and, likely, self-reported data as well. Analytics for this level of personalization in MOOCs have yet to be achieved but personalization is occurring even without adaptive algorithms, as distributed learners are primarily interfacing with content at their own machines, at their own pace. Finally, collective intelligence focuses on the vast informational network that is produced by and further enables the participatory, creative moments of the users of the Web [3]. Each individual learner in a MOOC enjoys a one-to-many style of communication that is enabled by discussion boards and other tools for peer-to-peer interaction. In the aggregate, this becomes many-to-many, a network of participants that can be tapped into or contributed to by any individual in order to share knowledge, give or get assistance with difficult problems, make sense of the expectations of faculty, or simply to experience and add to the social presence of the virtual experience.

These themes are embodied in a range of epistemological stances towards two core dimensions of learning environments: the location of knowledge and conceptions of the learner. Assessment is the third core dimension of the learning environment [4]. The technology enables a wide number of assessment types but the stances towards assessment follow not from the affordances of the Web but from the standard distinction between formative and summative assessments. However, instead of using this jargon, I choose language that reflects the nature of the interaction enabled by each type of assessment, as the central mechanism of learning in online settings are interactions among learners, resources, and instructors [5] Finally, it is important to note that this framework treats the instructor as a designer and an expert participant, which also leaves room for the expert role to be played by others such as teaching assistants.

***Knowledge*: Instructionist-participatory**
Where are opportunities to acquire or generate knowledge? Does knowledge live purely with the instructor and other expert participants or does it live in the broad universe of participants? Who has the authority to create and deliver content? Is the learning experience created solely by the course designers or is it co-created by learners?

***Learner*: Personalized-Collectivist**
Are learners cognitively and culturally unique beings, or members of a network? Do the learning opportunities in the course focus on the individual learner or on the interactions of the group?

***Assessment*: Evaluation-Feedback**

> What opportunities are provided for learners to make explicit their progress in knowledge construction? Are assessments designed to tell learners if they're right or to give them guidance for improvement?

The poles of each stance, as named above, are opposed to each other epistemologically, but one end is not necessarily preferable to the other. The choice between each stance is predicated on what is valued by the designer in a learning environment or learning experience, and what is known about effective instruction and learning activities from the learning sciences. Each feature of the course can be characterized along one or more of these dimensions (see Section 4.1). This means that multiple stances can exist in the same course.

3 A Proposed Taxonomy *(Grounding)*

The proposed taxonomy includes two levels of descriptive metadata. The first level characterizes course as a whole and is meant to evoke the broad set of opportunities available for sharing knowledge with MOOCs. The second level takes in turn each element of the interactive learning environment (ILE) and develops a list of possible features for the implementation of these elements, based on current and potential MOOC designs. The features on this level can also serve as a set of guidelines of options for course designers. In multiple iterations of the course, many of these fields will stay the same but others will change. Most fields will be limited to one tag but others could allow multiple (e.g. target audience in General Structure).

The architecture and options for metadata on learning objects has been a subject in the field for quite some time, as repositories for learning objects and OER have become more common. While I am somewhat remiss to throw yet another taxonomy into the mix, I believe that it is important to represent the unique role of MOOCs in an evolving ecosystem of lifelong learning opportunities. Because the content and structure of a MOOC is not limited by traditional institutional exigencies of limited seats or approval of a departmental committee and accreditation agencies, it becomes a vessel for knowledge sharing, competency development, and peer connections across all domains, from computer science to music production and performance.[2] As a technology it is agnostic to how it is used, which means that it can be designed in any way that our epistemological stances guide us to imagine. Education has goals ranging from knowledge development to civic participation and MOOCs can be explicitly designed to meet any of these goals.

3.1 General MOOC Structure

On the highest level, each MOOC needs to be characterized in terms of its subject matter, audience, and use. Table 1 presents the proposed categories and subcategories for the General MOOC Structure. With an eye towards future interoperability, where

[2] That said, there is an ongoing conversation about integrating MOOCs back into the pre-existing educational institutions, so the taxonomy must ne conversant with these efforts while also representing the vagaries of the moocspace as a separate ecosystem.

possible I use the terminology from the Learning Resources Metadata Initiative (LRMI) specification [7], or note in parentheses which LRMI field the moocspace categories could map onto.

Table 1. Categories and Subcategories for General MOOC Structure

- Name (LRMI)
- Numeric ID (auto-generated)
- Author (LRMI)
  - Faculty member
- Publisher (LRMI)
  - Affiliated university or other institution
- Platform
- inLanguage (LRMI)
  - primary language of resource
- Domain (*about*)
  - Computational /STEM – CS, math, science, computational social sciences, etc.
  - Humanist – humanities, non-computational social sciences, etc.
  - Professional – business, medicine, law, etc.
  - Personal – health, thinking, speaking, writing, art, music, etc.
- Level (*typicalAgeRange* or *educationalRole)*
  - Pre-collegiate; basic skills (i.e. gatekeeper courses, college/career-ready); undergraduate; graduate; professional development; life skills
- Target audience (*educationalRole*)
  - Current students, current professionals, lifelong learners
- Use (*educationalUse* or *educationalEvent*)
  - Public course (date(s) offered), content for "wrapped" in-person course (location and date(s) offered)
- Pace
  - Cohort-based vs. self-paced (*learningResourceType* or *interactivityType*)
  - Expected workload for full course (total hours, hours/week) (*timeRequired*)
- Accreditation
  - Certificate available
  - Transfer credit

3.2 Elements of the Interactive Learning Environment (ILE)

The ILE is made up of a set of learning objects, socio-technical affordances, and instructional and community design decisions. These features are created by the course designers -- instructors and technologists – and interpreted by learners throughout their ongoing interaction with the learning objects in the course, as well as the other individuals who are participating in the course (as peers or instructors).[3] The features of the ILE can be sorted into four distinct categories: instruction, content, assessment, and community. Table 2 lists out the possible features of the ILE, based on the current trends in MOOC design. As stated, this is a descriptive list - based on

[3] The individual- and group-level learning experiences that take place in the ILE are enabled by the technological infrastructure of the MOOC platform and mediated by learner backgrounds (e.g. prior knowledge, self-regulation and study habits) and intentions for enrolling [8] as well as the context in which the MOOC is being used (e.g. in a "flipped" classroom, with an informal study group, etc.). The relationship of these psychological and contextual factors to learning experiences and outcomes is a rich, multifaceted research area, which I put aside here to foreground the ILE and systematically describe the dimensions along which it varies.

the current generation of MOOCs – but will be expanded in the future, both to reflect new trends in MOOC design and to take a normative stance on potential design choices that are based in principles of the learning sciences or interface design. Some of the features are mutually exclusive (i.e. lecture types) but others could occur simultaneously in the same MOOC (i.e. homework structure). Most features will need to be identified by spending some time exploring the course, ideally while it is taking place.

Table 2. Features of ILE

Instruction • Lecture ▪ “traditional”: 1-3 hrs/wk, 20+ mins each ▪ “segmented”: 1-3 hrs/wk, 5-20 mins each ▪ “minimal”: <1 hr/wk • Readings • Simulations/inquiry environments/virtual labs • Instructor involvement – range from highly interactive to “just press play”	**Content** • Domain (in General Structure) • Modularized ▪ Within the course ▪ connected with other MOOCs/OER • Course pacing ▪ Self-paced ▪ Cohort-based
Assessment • In-video quizzes ▪ multiple choice vs. open-ended • Homework structure ▪ Multiple-choice ▪ Open-ended problems ▪ Performance assessments ▪ Writing assignments or programming assignments ▪ Videos, slides, multimedia artefacts • Group projects • Practice problems (non-credit bearing) ▪ Grading form–Quantitative, Qualitative • Grading structure (relevant to all credit-bearing assessments) ▪ Autograded ▪ Peer assessment, self-assessment, both ▪ Multiple submissions	**Community** • Discussion board • Social Media - Facebook group, Google+ community, twitter hashtag, reddit, LinkedIn, etc. • Blogs / student journals (inside or outside of platform) • Video chat (G+ hangout, Skype) • Text chat

4 The Taxonomy, Applied

4.1 Example of course mapping

Each course feature can be mapped onto one or more epistemological stances. The course overall can then be characterized by the overall epistemological tendencies of the course features. Table 3 provides an example.

Table 3. Mapping “Crash Course in Creativity” to the Taxonomy

General	Name: Crash Course in Creativity Author: Tina Seeling Publisher: Stanford Platform: NovoEd Domain: personal-thinking	Level: life skills Target audience: lifelong learners Use: public course (fall 2012) Pace: cohort-based - **collectivist** Certificate: yes
ILE and Stances		
Instruction	Lecture: minimal – 5-10 mins/wk to inspire group projects – **participatory** Readings: free, from her book - **instructionist**	
Content	Not modularized - **instructionist**	
Assessment	One individual creative projects – **participatory, individualist** Three group creative projects – **participatory, collectivist** Peer grading with qualitative comments–**participatory, feedback, collectivist**	
Community	Discussion board – **participatory, collectivist**	
OVERALL	**Participatory, collectivist, feedback**	

4.2 Stances to guide best practices and analytics.

The stances are not normative but do help specify which traditions of instructional and interface design should be turned to for guidance in best practices for designing resources. For example: instructionist lecture videos should follow the principles of multimedia learning, including balancing and integrating visual and verbal representations, relying on segmented (and learner-paced) narratives, and providing signaling mechanisms for the upcoming structure and content of a lecture. [9] The underlying epistemologies can also provide guidance about the type of analytics that are appropriate to for characterizing success in the design of the MOOC. For example, group-level outcomes may be more compelling for a collectivist MOOC – what is the overall level of interaction between learners, what kind of social networks form, with group projects can we characterize group composition or dynamics that lead to higher grades?

4.3 Centralizing distributed science: a short description of the moocspace

The taxonomy is a high-level, qualitative categorization of MOOCs that will allow for meaningful comparison across shared metrics about the courses. The taxonomy will be most usefully implemented in the *moocspace* – a digitized repository of knowledge about the research and production of massive open online courses – so named because it is an abstraction and reflection of the larger moocspace. The MOOC, abstracted, will be the central object of the moocspace, attached to standard metrics about the course, as well as reports on any research that has been done with data from that MOOC.[4] Variations in metrics could be related to aspect of the course

[4] Developing a small, meaningful set of shared metrics for MOOCs is currently an open question. Higher education in the US is characterized by enrollment rates at the beginning of

design, which are formalized in the taxonomy. Beyond descriptive data, a transparent, well-organized research base will enable an incremental and cumulative set of evidence from both exploratory studies (e.g. building learner models based on observational data) and experiments on the multiplicity of instructional and interface design features. A well-documented experiment in a small number of MOOCs could be replicated elsewhere by other researchers, and the findings could be synthesized by a third group by comparing results across variations in course features.

The moocspace could also be expanded to include the content of the MOOC itself, if licensing decisions are made that will allow MOOCs to become re-usable and re-mixable pieces of OER. This implementation would involve paradata on the uses of MOOC materials and incorporate a community aspect where faculty who use the materials could talk about what worked or didn't work in their courses. Finally, the MOOC object could also be attached to open datasets on MOOCs. The individuals who using such datasets may not be inside the academy, which underscores the need to build a structure for sharing newly developed knowledge back with the community.

If the moocspace is to be implemented, we will need develop consensus on the features in the taxonomy, as well as a strategy for tagging existing courses (crowdsourced? local experts?) and for adding new features to the taxonomy.

Acknowledgments. Much thanks to Kimberly Hayworth for her role in the initial brainstorming of MOOC features; thanks also to the Lytics Lab, Roy Pea, Zach Pardos, and the anonymous moocshop reviewers for feedback on earlier versions.

4 References

1. Jenkins, H. (2009) *Confronting the challenges of participatory culture: Media education for the 21st century*. Cambridge, MA: MIT Press.
2. National Educational Technology Plan. (2010). *Transforming American Education: Learning Powered by Technology.* Washington, DC: US Department of Education, Office of Educational Technology.
3. Lévy, P., & Bonomo, R. (1999). *Collective intelligence: Mankind's emerging world in cyberspace*. Perseus Publishing.
4. Brown, A. L., & Cocking, R. R. (2000). *How people learn*. J. D. Bransford (Ed.). Washington, DC: National Academy Press.
5. Anderson, T. "Towards a theory of online learning." (2004) *Theory and practice of online learning*: 3-31. Athabasca University, retrieved from http://cde.athabascau.ca/online_book/ch2.html
6. LRMI Specification Version 1.1 (Apr 28, 2013). www.lrmi.net/the-specification
7. Grover, S., Franz, P., Schneider, E. and Pea, R. (2013) "The MOOC as Distributed Intelligence: Dimensions of a Framework for the Design and Evaluation of MOOCs." In *Proceedings of the 10th International Conference on Computer Supported Collaborative Learning* (Madison, WI, June 16-19).
8. Mayer, R E., ed. (2005) *The Cambridge handbook of multimedia learning*. Cambridge University Press.

the semester, and persistence rates and completion rates over time. In addition to enrollment and activity rates initially and over time, for open courses it may be more appropriate to examine levels of engagement, time-on-task, or participation on the discussion forum.

Roll Call: Taking a Census of MOOC Students

Betsy Williams[1],

[1] Stanford University, Graduate School of Education, 520 Galvez Mall,
CERAS Building, 5th Floor, Stanford, CA, 94305, USA
{betsyw@stanford.edu}

Abstract. This paper argues for spending resources on taking a high quality census or representative survey of students on who enroll with all major MOOC platforms. Expanded knowledge of current students would be useful for business and planning, instruction, and research. Potential concerns of cost, privacy, stereotype threat, and maladaptive use of the information are discussed.

Keywords: MOOC population, education, data collection, survey, demography

1 Introduction

Quantitative education researchers are accustomed to piecing together complex analyses from the rather lifeless data available from administrative records and test scores. The fine-grained data collected by MOOCs—including detailed knowledge of students' attendance and attention patterns, response on formative and summative assessments, and discussions with instructors and fellow students—offer an opportunity for much greater understanding of teaching and learning.

Unfortunately, MOOCs are not making the most out of their big data because they are not collecting enough data on students' backgrounds. Borrowing Bayesian terms, platforms have few priors on students, even though these priors can have great predictive power if paired with existing knowledge, from fields like developmental psychology and higher education theory.

The major platforms optimize sign up to make becoming part of the platform as quick as possible, leaving students mostly mysterious. EdX requests a few valuable pieces of demographic data upon registration, asking for voluntary identification by gender, year of birth, level of education completed, and mailing address without a clear reason why.[1] Coursera's information gathering is more like social media or a dating site, encouraging students who visit the profile page to share their age, sex, and location. As part of its "About Me" prompt, Coursera suggests that among other things users might share "what you hope to get out of your classes," while EdX asks the question in an open-ended text box upon registration. While these questions yield some of the data that is valuable for improving courses, the platforms, and education

[1] No one reads terms of service [1].

research, I argue that the platforms should collect more key data, clearly identified as information that will not be sold or used for targeted marketing or for student evaluation.

The paper first describes the fields most useful for analysis based on priors, and then it explains the benefits to platform development, instructional quality, and research. Potential drawbacks are discussed, including cost, privacy concerns, the risk of invoking stereotype threat, and the potential for undesirable changes to arise from this information.

2 Prior Information about Students

Given infinite data storage and infinite indulgence on the part of MOOC students, knowing every scrap of data about students might allow for inspired analyses and eerily predictive machine learning exercises. However, a more humble conception of student data would ably fulfill our research needs.

Core demographic information includes year of birth, gender, and race/ethnicity.[2] Asking users for their current city or place of residence should generate more accurate location results than IP address tracing or the information provided to appear on a semi-public profile. Combined with place of origin and native language, these questions provide a sketch of a student's likely history and culture.

A MOOC-run survey would also provide the opportunity to ask questions less often available in administrative education data, although extremely useful for understanding who enrolls. Although sensitive, questions about socioeconomic status and living situation would be tremendously helpful; for instance, is a student living with family, and to which generation does that student belong?

Adult students' lives are increasingly complex, and questions about work and education history should do their best to capture this. If a student's highest degree is a high school diploma (or equivalent), then have they ever enrolled in higher education? If so, in how many institutions? How many years and months would they estimate this spanned? Were they primarily taking full time or part time loads? What was the name of their primary institution, and what was their most recent course of study pursued? Those who have earned bachelor's degrees or higher should face similar questions. For all students, questions about previous or concurrent MOOC use would be very valuable. Work history can get a similar treatment, identifying such things as area of employment, and full- and part-time scheduling.

Although students themselves may not be entirely clear on the point yet, questions about educational and career goals, along with goals for the course, would be extremely useful. This information is captured to some extent in existing questions or for particular research. However, this may be incomplete or collected only in a piecemeal fashion. For instance, a study on learner patterns surveys the students in

[2] Race and ethnicity are social constructs whose meaning greatly varies by national context. For instance, being white in Norway has a different social meaning than being white in South Africa. And Belgium is split by a key ethnic marker—Walloon versus Fleming—that does not matter in other countries. Thus, choices for race/ethnicity should be based on the selected country of origin and/or country of residence.

one course, asking for intentions in the course, current employment status, years of work history, and highest degree attained [2].

Valuable information from surveys need not all be based on recall or opinion. Meaningful priors about academic preparation in particular fields can be generated by computer adaptive test questions in key content domains, based on existing work in psychometrics. Behavioral economics shows that survey questions can measure levels of risk aversion (asking for preferences between a gamble for $X and receiving $Y with certainty) and time discounting on money (asking about preferences for receiving $X now or $Y at a certain point in the future).[3]

Finally, there's a useful realm of information about how students use the platform. Within a class, how much time do they plan to devote, how do they plan to interact with peers, and will they use external supports, such as tutors, websites, and textbooks? What modes of access to the course are available to them? In particular, what electronic devices are available to them, is their use of the devices limited, and what kind of Internet access is available?

3 Value for Planning and Strategy

The background information on users discussed above provides extremely valuable data for the operations of the course platforms. Let us stipulate that there are limitations on the data being used for targeted marketing purposes. Even so, having aggregate background information on who is using which MOOCs is a huge advance.

In a traditional business mindset, the primary questions would be who is willing and who is able to pay. However, more advanced uses could help a course recommendation engine distinguish who is taking the course as a consumption good versus as investment in their future; the follow-up courses the students are interested in may be vastly different.

The survey may also suggest a greater than anticipated demand for classes taught at a certain level or on a certain topic. Students' locations, educations, and work histories might help the platform identify other institutions that may be good partners, either because they are very well-represented or under-represented.

4 Value for Instructional Design

A strong finding in educational research is that there is not a single correct way to teach or structure a course. Instead, learners matter, and knowledge about the students and their characteristics is important for teaching well [3]. Knowing more about the students also allows instructors to effectively call on their existing knowledge and address likely misconceptions; this is part of Pedagogical Content Knowledge [3] and a prominent contribution of Piagetian constructivism [4].

[3] For the most accurate answers, survey takers would actually receive the payout they say they prefer, subject to a gamble or delay as the case may be.

For example, knowing the age distribution and native languages of students can improve instruction. Instructors may choose allusions, words, and examples better.

An inherent challenge within the online classroom is that some feedback that is obvious in a physical context is not available. One student falling asleep in a lecture hall is far more obvious and effective of a signal than a thousand who never rewind the recorded lectures. While learning analytics is tackling this paucity of data in clever ways, we would also benefit greatly from leaning on priors. Imagine two students who do not watch the second week lecture by the beginning of week three: one has a doctoral degree in the field, while the other is a high school graduate who has attended several different institutions of higher education and intends to take a course for professional development. Applying theory to this prior knowledge, we might think the former finds the course matter unnecessary to review, while the latter may be struggling to stay motivated in the class.

In short, better prior knowledge can be paired with data collected in courses to better identify how students are learning the course content and improve the course.

5 Value for Education Research

MOOC populations are so wildly self-selected, and the field so new, that external validity is extremely questionable. At best, we might extend findings in a class to perhaps the same class the next time it is taught or use the results to develop hypotheses and learning theory.

While there is great value in using research to improve a single course, ideally the lessons could be transferred more broadly, so that the effort of analysis pays greater dividends. However, results cannot generalize until the population of the study is understood; once more is known about incoming characteristics of MOOC students who were studied, researchers can seek other classes that resemble them in salient details.

More concretely, MOOCs offer radical levels of access to education, and so they include many non-traditional and out-of-school learners. These nontraditional learners can be elusive research subjects, and there is also great diversity among their numbers. Having additional background data allows us to tag them and better understand their behavior. If a course platform is successful with a particular college level course and is contemplating recommending it to a partner community college, it would be wise to understand how students of different backgrounds performed. The inference is not direct, but it is far more useful than a recommendation based on coarser data.

The MOOC is also a fantastic platform for learning about how everyone learns, not just how self-selected MOOC users learn. The large number of students and the computerized means of instruction mean that MOOCs are very amenable to experimentation and careful observation. In addition, the very design of MOOCs strips down the traditional classroom; greater insight about learning and traditional instruction can come from adding back in some of these elements that are taken for granted in other classrooms.

Yet again, the great advantages of MOOCs as a place for learning research have the caveat that results are hard to generalize. However, if researchers control for the observable background data of the students who opt into MOOCs, their results will be far more plausibly applicable to a wide array of classes.

A key challenge within the online classroom is that feedback that may be obvious in a physical context, such as real-time indications of student engagement or confusion, is usually not available. One student falling asleep in a lecture hall is far more obvious and effective of a signal than a thousand who never rewind the recorded lectures. While learning analytics is tackling this paucity of data in clever ways, we would also benefit greatly from leaning on priors. Imagine two students who do not watch the second week lecture by the beginning of week three: one has a doctoral degree in the field, while the other is a high school graduate who has attended several different institutions of higher education and intends to take a course for professional development. Applying theory to this prior knowledge, we might think the former finds the course matter unnecessary to review, while the latter may be struggling to stay motivated in the class.

In short, better prior knowledge can be paired with data collected in courses to better identify how students are learning the course content.

6 Concerns and Limitations

There are genuine concerns with collecting this much data. Here, I discuss cost, privacy, stereotype threat, and maladaptive use. I present these cursorily not to dismiss these points, but to begin what must be a larger discussion.

6.1 Cost

Course platforms are in a unique position It can be extremely costly to ask survey questions. User attention is limited and a choice to ask an additional question may implicitly limit their engagement later during the session, or even drive them away from the service at the extreme. Higher quality survey data can be generated by using internal resources to follow up with non-responders; higher response rates can also be generated by incentives, such as monetary payment, entry in a lottery, or access to a premium site feature. In addition, comprehensive surveys offered by a platform itself can be more easily embedded in the site, making the survey more available and more salient.

Administering a vast survey at the site level also better captures students who might be over-sampled if asked class-by-class. Cross-course analyses can be conducted more easily if the relatively permanent, detailed background information is available at the platform level, rather than asked for in individual courses.

Stratified sampling methods could be used to reduce the burden on students and the cost burden on the platform. For instance, core questions could be asked of the main sample of students, while additional long forms of the survey ask different questions of different students. The aggregate picture can be pieced together with a smaller burden on most students and a lower cost to the platform. While this is less than ideal, it may be a necessary tradeoff in some cases.

6.2 Privacy

Privacy concerns are important and complex, and researchers are used to the question of balancing privacy concerns against the benefit of the research. The more background data a platform collects, the more risk that personally identifiable information about subjects is available through composite reports or if the data are intercepted. Access security and care in reporting results are thus crucial and should be considered ahead of time.

Because of these concerns or others, some students may wish not to provide information, which could systematically bias the survey sample, making our inferences worse. Some students who wish to opt out may be reassured if the reasons for the research and the protection of the data are made clear. Others may be more comfortable with anonymized options for responding or techniques designed for collecting sensitive data. [5]

6.3 Stereotype Threat and Maladaptive Use of Information

Arguably, the Internet provides one of the few places in society where people are not forced to reveal information about their social position, which may be of value in itself.[4] A powerful strand of research in social psychology suggests that invoking identities that are attached to negative stereotypes can hinder educational performance; people are especially vulnerable to this "stereotype threat" if they feel there is a power imbalance and that they are being defined by others' judgments [8]. This threat could both change answers provided and potentially harm the student. However, a sustained harm seems unlikely to result from the trigger of a few questions on a survey; rather, the underlying negative social context or vulnerability might be in play. It would be unfortunate if a detailed survey triggered stereotype threat, even temporarily, but making sure the questions are seen as low-stakes could help.

There may also be a risk that instructors change their courses in unintended ways if they find out more about the students. An instructor might make a college-level course less rigorous if he finds out high school students are enrolled, for instance. While this raises concerns, it is ultimately up to policy and instructors' judgment.

[4] Perhaps the Internet is the place where students "will not be judged by the color of their skin, but by the content of their character." [6] Less seriously, "On the Internet, nobody knows you're a dog." [7]

7 Conclusion

Platform operations, instructional design, and educational research would all benefit from collecting more systematic background data about students. Better knowledge about who takes MOOCs is crucial at this stage in their lifetime. I propose not only a census of MOOC users on each platform, capturing a snapshot of users today, but an ongoing effort to capture these detailed demographic snapshots at least every three years.

Acknowledgments. Many thanks to Eric Bettinger, Susanna Loeb, Tom Dee, Roy Pea, Mitchell Stevens, and participants in the Lytics Lab for valuable discussions that have influenced this paper.

References

1. Gindin, S.E.: Nobody Reads Your Privacy Policy or Online Contract? Lessons Learned and Questions Raised by the FTC's Action Against Sears. 8 Nw. J. Tech & Intell. Prop. 1, 1--38 (2009)
2. Kizilcec, R.F, Piech, C., Schneider, E.: Deconstructing Disengagement: Analyzing Learner Subpopulations in Massive Open Online Courses. In: 3rd Conference on Learning Analytics and Knowledge. Leuven, Belgium (2013)
3. Shulman, L.S.: Knowledge and Teaching: Foundations of the New Reform. Harvard Educational Rev. 57, 1--22 (1987)
4. Ackermann, E.K.: Constructing Knowledge and Transforming the World. In: Tokoro, M., Steels, L. (eds.) A Learning Zone of One's Own: Sharing Representations and Flow in Collaborative Learning Environments. pp. 15--37. IOS Press, Amsterdam, Berlin, Oxford, Tokyo, Washington, DC (2004)
5. Du, W., Zhan, Z.: Using Randomized Response Techniques for Privacy-Preserving Data Mining. In: 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 505-510. ACM, New York (2003)
6. King, M.L.: I Have a Dream. In: Carson, C., Shepard, K. (eds.) A Call to Conscience: The Landmark Speeches of Dr. Martin Luther King, Jr. IPM/Warner Books, New York (2001)
7. Steiner, P.: On the Internet, Nobody Knows You're a Dog. The New Yorker LXIX, 20, p. 61 (1993)
8. Walton, G.M., Paunesku, D., Dweck, C.S.: Expandable Selves. In: Leary, M.R., Tangney, J.P. (eds.) The Handbook of Self and Identity, Second Edition, pp. 141--154. Taylor and Francis, New York (2012)

MOOCdb: Developing Data Standards for MOOC Data Science

Kalyan Veeramachaneni, Franck Dernoncourt,
Colin Taylor, Zachary Pardos, and Una-May O'Reilly

Massachusetts Institute of Technology, USA.
{kalyan,francky,colin_t,zp,unamay}@csail.mit.edu

1 Introduction

Our team has been conducting research related to mining information, building models, and interpreting data from the inaugural course offered by edX, *6.002x: Circuits and Electronics*, since the Fall of 2012. This involves a set of steps, undertaken in most data science studies, which entails positing a hypothesis, assembling data and features (aka properties, covariates, explanatory variables, decision variables), identifying response variables, building a statistical model then validating, inspecting and interpreting the model. In our domain, and others like it that require behavioral analyses of an online setting, a great majority of the effort (in our case approximately 70%) is spent assembling the data and formulating the features, while, rather ironically, the model building exercise takes relatively less time. As we advance to analyzing cross-course data, it has become apparent that our algorithms which deal with data assembly and feature engineering lack cross-course generality. This is not a fault of our software design. The lack of generality reflects the diverse, ad hoc data schemas we have adopted for each course. These schemas partially result because some of the courses are being offered for the first time and it is the first time behavioral data has been collected. As well, they arise from initial investigations taking a local perspective on each course rather than a global one extending across multiple courses.

In this position paper, we advocate harmonizing and unifying disparate "raw" data formats by establishing an open-ended standard data description to be adopted by the entire education science MOOC oriented community. The concept requires a schema and an encompassing standard which avoid any assumption of data sharing. It needs to support a means of sharing *how the data is extracted, conditioned and analyzed.*

Sharing scripts which prepare data for models, rather than data itself, will not only help mitigate privacy concerns but it will also provide a means of facilitating intra and inter-platform collaboration. For example, two researchers, one with data from a MOOC course on one platform and another with data from another platform, should be able to decide upon a set of variables, share scripts that can extract them, each independently derive results on their own data, and then compare and iterate to reach conclusions that are cross-platform as well as cross-course. In a practical sense, our goal is a standard facilitating insights

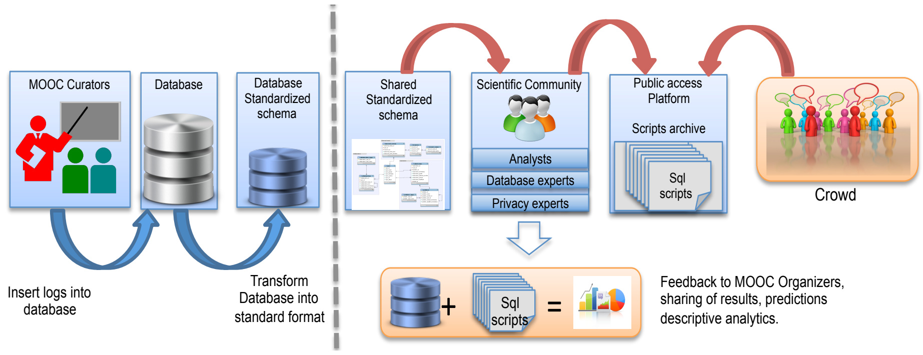

MOOC Curators
Database
Database Standardized schema
Insert logs into database
Transform Database into standard format
Shared Standardized schema
Scientific Community
Analysts
Database experts
Privacy experts
Public access Platform
Scripts archive
Sql scripts
Crowd
+
Sql scripts
=
Feedback to MOOC Organizers, sharing of results, predictions descriptive analytics.

Fig. 1. This flowchart represents the context of a standardized database schema. From left to right: Curators of MOOC course format the raw transaction logs into the schema and populate either private or public databases. This raw database is transformed into a standard schema accepted by the community, (like the one proposed in this paper) and is exposed to the analytics community, mostly researchers, who develop and share scripts, based upon it. The scripts are capable of extracting study data from any schema-based database, visualizing it, conditioning it into model variables and/or otherwise examining it. The schema is unifying while the scripts are the vehicle for cross-institution research collaboration and direct experimental comparison.

from data being shared without data being exchanged. It will also enable research authors to release a method for recreating the variables they report using in their published experiments.

Our contention is that the MOOC data mining community - from all branches of educational research, should act immediately to engage in consensus driven discussions toward a means of standardizing data schema and building technology enablers for collaborating on data science via sharing scripts, results in a practical, directly comparable and reproducible way. It is important to take initial steps now. We have the timely opportunity to avoid the data integration chaos that has arisen in fields like health care where large legacy data, complex government regulations and personal privacy concerns are starting to thwart scientific progress and stymy access to data. In this contribution, we propose a standardized, cross-course, cross-platform, database schema which we name as *"MOOCdb"*. [1]

We proceed by describing our concept and what it offers in more detail in Section 2. Section 3 details our proposed the data schema systematically. Section 4 shows, with a use case, how the schema is expressive, supportive and reusable. Section 5 concludes and introduces our current work.

[1] We would like to use the MOOCshop as a venue for introducing it and offering it up for discussion and feedback. We also hope to enlist like minded researchers willing to work on moving the concept forward in an organized fashion, with plenty of community engagement.

2 Our Concept and What it Offers

Our concept is described as follows, and as per Figure 1:

- It identifies two kinds of primary actors in the MOOC eco-system: *curators* and *analysts*. Curators collect raw behavioral data expressing MOOC students' interaction with online course material and then transfer it to a database, often as course content providers or course platform providers. *Analysts* reference the data to examine it for descriptive, inferential or predictive insights. The role of the analysts is to visualize, descriptively analyze, use machine learning or otherwise interpret some set of data within the database. Analysts extract, condition (e.g. impute missing values, de-noise), and create higher level variables for modeling and other purposes from the data. To perform this analysis, they first transform the data into the standard schema and compose *scripts* or use publicly available scripts when it suffices. They also contribute their scripts to the archive so others can use.
- It identifies two types of secondary actors: the *crowd*, and the data science experts (database experts and privacy experts). When needs arise, the community can seek the help of the *crowd* in innovative ways. Experts contribute to the community by providing state-of-the art technological tools and methods.
- A common standardized and shared schema into which the data is stored. The schema is agreed upon by the community, generalizes across platforms and preserves all the information needed for data science and analytics.
- A shared community-oriented repository of data extraction, feature engineering, and analytics scripts.
- Over time the repository and the schema, both open ended, grow.

This concept offers the following:
The benefits of standardization: The data schema standardization implies that the raw data from every course offering will be formatted the same way in its database. It ranges from simple conventions like storing event timestamps in the same format to common tables, fields in the tables, and relational links between different tables. It implies compiling a scientific version of the database schema that contains important events, fields, and dictionaries with highly structured data is amenable for scientific discovery. Standardization supports cross-platform collaborations, sharing query scripts, and the definition of variables which can be derived in exactly the same way for irrespective of which MOOC database they come from.
Concise data storage: Our proposed schema is "loss-less", i.e. no information is lost in translating raw data to it. However, the use of multiple related tables provides more efficient storage.
Savings in effort: A schema speeds up database population by eliminating the steps where a schema is designed. Investigating a dataset using one or more existing scripts helps speed up research.
Sharing of data extraction scripts: Scripts for data extraction and descriptive statistics extraction will be open source and can be shared by everyone.

Some of these scripts could be very general and widely applicable, for example: ”For every video component, provide the distribution of time spent by each student watching it?” and some would be specific for a research question, for example generation of data for Bayesian knowledge tracing on the problem responses. These scripts could be optimized by the community and updated from time to time.

Crowd source potential: Machine learning frequently involves humans identifying explanatory variables that could drive a response. Enabling the crowd to help propose variables could greatly scale the community’s progress in mining MOOC data. We intentionally consider the data schema to be independent of the data itself so that people at large, when shown the schema, optional prototypical synthetic data and a problem, can posit an explanatory variable, write a script, test it with the prototypical data and submit it to an analyst. The analyst can assess the information content in the variable with regards to the problem at hand and rank and feed it back to the crowd, eventually incorporating highly rated variables into learning.

A unified description for external experts: For experts from external fields like“Very Large Databases/Big Data” or ”Data Privacy”, standardization presents data science in education as unified. This allows theme to technically assist us with techniques such as new database efficiencies or privacy protection methods.

Sharing and reproducing the results: When they publish research, analysts share the scripts by depositing them into a public archive where they are retrievable and cross-referenced to their donor and publication.

Our concept presents the following challenges:

Schema adequacy: A standardized schema must capture all the information contained in the raw data. To date, we have only verified our proposed schema serves the course we investigated. We expect the schema to significantly change as more courses and offerings are explored. It will be challenging to keep the schema open ended but not verbose. While a committee could periodically revisit the schema, a more robust approach would be to let it evolve through open access to extension definitions then selection of good extensions via adoption frequency. This would embrace the diversity and current experimental nature of MOOC science and avoid standard-based limitations. One example of a context similar to the growth of MOOCs is the growth of the internet. HTML and Web3.0 did not rein in the startling growth or diversity of world wide web components. Instead, HTML (and its successors and variants) played a key role in delivering content in a standardized way for any browser. The semantic web provides a flexible, community driven, means of standards adoption rather than completely dictating static, monolithic standards. We think there are many lessons to learn from the W3C initiative. To whit, while we provide ideas for standards below, we propose that, more importantly, there is a general means of defining standards that allow interoperability, which should arise from the examples we are proposing.

Platform Support: The community needs a website defining the standard data template and a platform assisting researchers in sharing scripts. It requires tests for validating scripts, metrics to evaluate new scripts and an repository of scripts with efficient means of indexing and retrieval.

Motivating the crowd: How can we encourage large scale script composition and sharing so the crowd will supply explanatory variables? How can we provide useful feedback when the crowd is not given the data? KAGGLE provides a framework from which we can draw inspiration, but it fundamentally differs from what we are proposing here. KAGGLE provides a problem definition, a dataset that goes along with it, whereas we are proposing that we share the schema, propose a problem, give an example of a set of indicators and the scripts that enabled their extraction, and encourage users to posit indicators and submit scripts. Such an endeavor requires us to: define metrics for evaluation of indicators/features given the problem, provide synthetic data (under the data schema) to allow the crowd to test and debug their feature engineering scripts, and possibly visualizations of the features or aggregates over their features (when possible), and most importantly a dedicated compute resource that will perform machine learning and evaluate the information content in the indicators.

3 Schema description

We surveyed a typical set of courses from Coursera and edX. We noticed three different modes in which students engage with the material. Students observe the material by accessing all types of resources. In the second mode they submit material for evaluation and feedback. This includes problem check-ins for lecture exercises, homework and exams. The third mode is in which they collaborate with each other. This includes posting on forums and editing the wiki. It could in future include more collaborative frameworks like group projects. Based on these three we divide the database schema into three different tables. We name these three modes as *observing*, *submitting* and *collaborating*. We now present the data schema for each mode capturing all the information in the raw data.

3.1 The observing mode

In this mode, students simply browse and observe a variety of resources available on the website. These include the *wiki*, *forums*, *lecture videos*, *book*, *tutorials*. Each unique resource is usually identifiable by a *URL*. We propose that data pertaining to the observing mode can be formatted in a 5-tuple table: *u_id* (*user id*), *r_id* (*resource id*), *timestamp*, *type_id*, *duration*. Each row corresponds to one click event pertaining to student. Two tables that form the dictionaries accompany this event table. The first one maps each unique *url* to *r_id* and the second one maps *type_id* to resource type, i.e., *book*, *wiki*. Splitting the tables into event and dictionary tables allows us to reduce the sizes of the tables significantly. Figure 4 shows the schema and the links.

3.2 The submitting mode

Similar to the table pertaining to the observing mode of the student, we now present a structured representation of the problem components of the course.

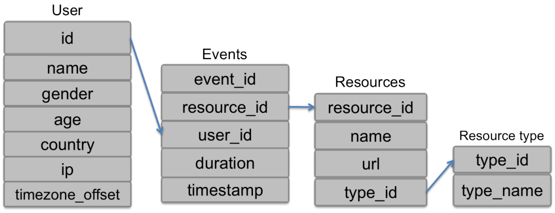


Fig. 2. Data schema for the observing mode

A typical MOOC consists of assignments, exams, quizzes, exercises in between lectures, labs (for engineering and computer science). Unlike campus based education, students are allowed to submit answers and check them multiple times. Questions can be multiple choice or a student can submit an analytical answer or even a program or an essay. Assessments are done by computer or by peers to evaluate the submissions [1]. We propose the following components:

Submissions table: In this table each submission made by a student is recorded. The 5 tuple recorded is *u_id*, *p_id*, timestamp, the answer, and the attempt number.

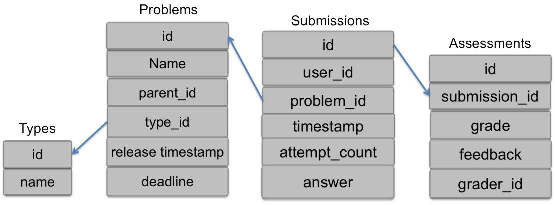


Fig. 3. Data schema for the submitting mode.

Assessments table: To allow for multiple assessments this table is created separately from the submissions table. In this table each assessment for each submission is stored as a separate row. This separate table allows us to reduce the size since we do not repeat the *u_id* and *p_id* for each assessment.

Problems table: This table stores the information about the problems. We *id* the smallest problem in the entire course. The second field provides the name for the problem. The problem is identified if it is a sub problem within another problem by having a parent *id*. Parent *id* is a reflective field in that its entries are one of the problem *id* itself. Problem type *id* stores the information about whether it is a homework, exercise, midterm or final. The table also stores the problem release date and the problem submission deadline date as two fields. Another table stores the id for problem types.

3.3 The Collaborating mode

Student interact and collaborate among themselves throughout the course duration through forums and wiki. In forums a student either initiates a new thread or responds to an existing thread. Additionally students can up vote, and down vote the answers from other students. In wiki students edit, add, delete and initiate a new topic. To capture this data we form the following tables with the following fields:

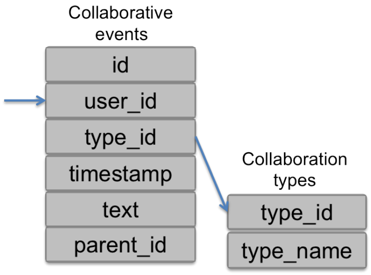


Fig. 4. Data schema for collaborating mode

Collaborations table: In this table each attempt made by a student to collaborate is given an *id*. The 5 fields in this table are *u_id*, collaboration type (whether wiki or forum), timestamp, the pointer to the text inserted by this user, and the parent *id*. The last field is a reflective field as well.

Collaboration type table: In this table the collaboration type *id* is identified with a name as to whether it is a wiki or a forum.

4 The edX 6.002x case study

edX offered its first course *6.002x: Circuits and Electronics* in the Fall of 2012. 6.002x had 154,763 registrants. Of these, 69,221 people looked at the first problem set, and 26,349 earned at least one point on it. 13,569 people looked at the midterm while it was still open, 10,547 people got at least one point on the midterm, and 9,318 people got a passing score on the midterm. 10,262 people looked at the final exam while it was still open, 8,240 people got at least one point on the final exam, and 5,800 people got a passing score on the final exam. Finally, after completing 14 weeks of study, 7,157 people earned the first certificate awarded by MITx, showing that they successfully completed 6.002x.

The data corresponding to the behavior of the students was stored in multiple different formats and was provided to us. These original data pertaining to the observing mode was stored in files and when we transcribed in the database with fields corresponding to the names in the "*name-value*" it was about the size of around 70 GB. We imported the data into a database with the schema we described in the previous subsections. The import scripts we had to build fell into two main categories:

- reference generators, which build tables listing every user, resource and problem that were mentioned in the original data.
- table populators, which populate different tables by finding the right information and converting it if needed.

The sizes and the format of the resulting tables is as follows: submissions: 341 MB (6,313,050 rows); events: 6,120 MB (132,286,335 rows); problems: 0.08 MB; resources: 0.4 MB; resource types: 0.001 MB; users: 3MB. We therefore reduced the original data size by a factor of 10 while keeping most of the information. This allows us to retrieve easily and quickly information on the students' activities. For example, if we need to know what is the average number of pages in the book a student read, it would be around 10 times faster. Also, the relative small size of the tables in this format allows us to do all the work in memory on any relatively recent desktop computer. For more details about the analytics we performed as well as the entire database schema we refer the reader to [2] [2]

5 Conclusions and future work

In this paper, we proposed a standardized data schema and believe that this would be a powerful enabler for ours and others researchers involved in MOOC data science research. Currently, we after building databases based on this schema we are developing a number of analytic scripts that extract multiple attributes for a course. We intend to release them in the near future. We believe it is timely to envision an open data schema for MOOC data science research.

Finally, we propose that as a community we should come up with a shared standard set of features that could be extracted across courses and across platforms. The schema facilities sharing and re-use of scripts. We call this the "feature foundry". In the short term we propose that this list is an open, living handbook available in a shared mode to allow addition and modification. It can be implemented as a google doc modified by the MOOC community. At the moocshop we would like to start synthesizing a more comprehensive set of features and developing the handbook. Feature engineering is a complex, human intuition driven endeavor and building this handbook and evolving this over years will be particularly helpful.

References

1. Piech, C., Huang, J., Chen, Z., Do, C., Ng, A., Koller, D.: Tuned models of peer assessment in MOOCs. In: Proceedings of The 6th International Conference on Educational Data Mining (EDM 2013). (2013)
2. Dernoncourt, F., Veeramachaneni, K., Taylor, C., O'Reilly, U.M.: Methods and tools for analysis of data from MOOCs: edx 6.002x case study. In: Technical Report, MIT. (2013)

[2] For the full MOOCdb database schema, see `http://bit.ly/MOOCdb`)

Syntactic and Functional Variability of a Million Code Submissions in a Machine Learning MOOC

Jonathan Huang, Chris Piech, Andy Nguyen, and Leonidas Guibas

Stanford University

Abstract. In the first offering of Stanford's Machine Learning Massive Open-Access Online Course (MOOC) there were over a million programming submissions to 42 assignments — a dense sampling of the range of possible solutions. In this paper we map out the syntax and functional similarity of the submissions in order to explore the variation in solutions. While there was a massive number of submissions, there is a much smaller set of unique approaches. This redundancy in student solutions can be leveraged to "force multiply" teacher feedback.

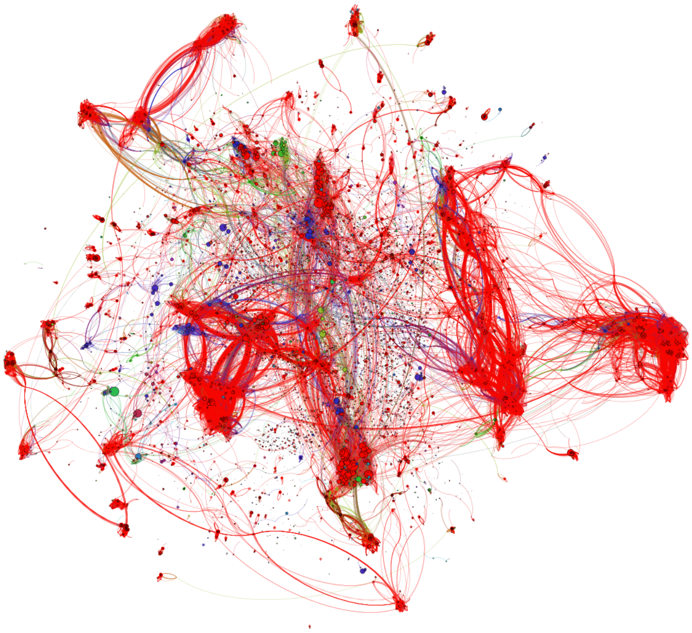


Fig. 1. The landscape of solutions for "gradient descent for linear regression" representing over 40,000 student code submissions with edges drawn between syntactically similar submissions and colors corresponding to performance on a battery of unit tests (red submissions passed all unit tests).

1 Introduction

Teachers have historically been faced with a difficult decision on how much personalized feedback to provide students on open-ended homework submissions such as mathematical proofs, computer programs or essays. On one hand, feedback is a cornerstone of the educational experience which enables students to learn from their mistakes. On the other hand, giving comments to each student can be an overwhelming time commitment [4]. In contemporary MOOCs, characterized by enrollments of tens of thousands of students, the cost of providing informative feedback makes individual comments unfeasible.

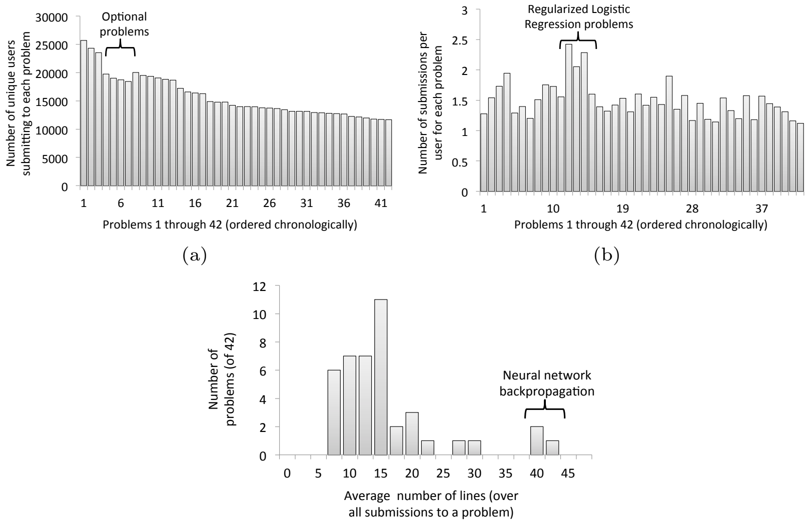
Interestingly, a potential solution to the high cost of giving feedback in massive classes is highlighted by the volume of student work. For certain assignment types, most feedback work is redundant given sufficiently many students. For example, in an introductory programming exercise many homework submissions are similar to each other and while there may be a massive number of submissions, there is a much smaller variance in the content of those submissions. It is even possible that with enough students, the entire space of reasonable solutions is covered by a subset of student work. We believe that if we can organize the space of solutions for an assignment along underlying patterns we should be able to "force multiply" the feedback work provided by a teacher so that they can provide comments for many thousands of students with minimal effort.

Towards the goal of force multiplying teacher feedback, we explore variations in homework solutions for Stanford's Machine Learning MOOC that was taught in Fall of 2011 by Andrew Ng (ML Class), one of the first MOOCs taught. Our dataset consists of over a million student coding submissions, making it one of the largest of its kind to have been studied. By virtue of its size and the fact that it constitutes a fairly dense sampling of the possible space of solutions to homework problems, this dataset affords us a unique opportunity to study the variance of student solutions. In our research, we first separate the problem of providing feedback into two dimensions: giving output based feedback (comments on the functional result of a student's program) and syntax based feedback (comments on the stylistic structure of the student's program). We then explore the utility and limitations of a "vanilla" approach where a teacher provides feedback only on the k most common submissions. Finally we outline the potential for an algorithm which propagates feedback on the entire network of syntax and output similarities. Though we focus on the ML Class, we designed our methods to be agnostic to both programming language, and course content.

Our research builds on a rich history of work into finding similarity between programming assignments. In previous studies researchers have used program similarity metrics to identify plagiarism [1], provide suggestions to students' faced with low level programming problems [2] and finding trajectories of student solutions [3]. Though the similarity techniques that we use are rooted in previous work, the application of similarity to map out a full, massive class is novel.

2 ML Class by the numbers

When the ML Class opened in October 2011 over 120,000 students registered. Of those students 25,839 submitted at least one assignment, and 10,405 submitted solutions to all 8 homework assignments (each assignment had multiple parts



(a)

(b)

(c)

Fig. 2. (a) Number of submitting users for each problem; (b) Number of submissions per user for each problem; (c) Histogram over the 42 problems of average submission line counts.

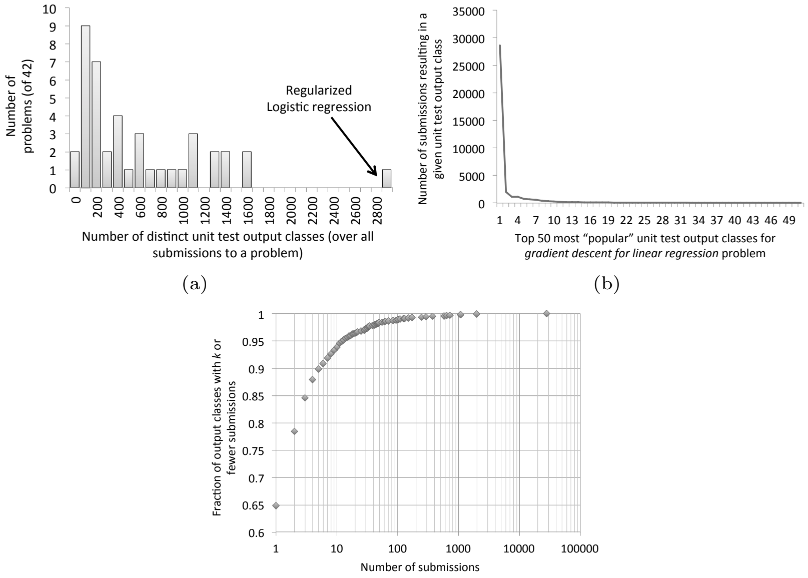
which combined for a total of 42 coding based problems) in which students were asked to program a short matlab/octave function. These homeworks covered topics such as regression, neural networks, support vector machines, among other topics. Submissions were assessed via a battery of unit tests where the student programs were run with standard input and assessed on whether they produced the correct output. The course website provided immediate confirmation as to whether a submission was correct or not and users were able to optionally re-submit after a short time window.

Figure 2(a) plots the number of users who submitted code for each of the 42 coding problems. Similarly, Figure 2(b) plots the average number of submissions per student on each problem and reflects to some degree its difficulty.

In total there were 1,008,764 code submissions with typical submissions being quite short — on average a submission was 16.44 lines long (after removing comments and other unnecessary whitespace). Figure 2(c) plots a histogram of the average line count for each of the 42 assignments. There were three longer problems — all relating to the backpropagation algorithm for neural networks.

3 Functional variability of code submissions

First, we examine the collection of unit test outputs for each submitted assignment (which we use as a proxy for *functional variability*). In the ML Class, the



(a)

(b)

(c)

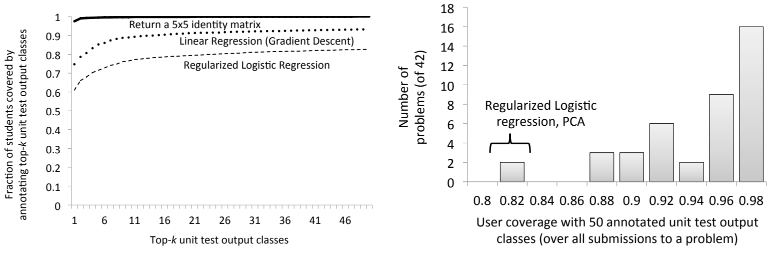
Fig. 3. (a) Histogram over the 42 problems of the number of distinct unit test outputs; (b) Number of submissions to each of the 50 most common unit test outputs for the "gradient descent for linear regression" problem; (c) Fraction of distinct unit test outputs with k or fewer submissions. For example, about 95% of unit test outputs owned fewer than 10 submissions.

unit test outputs for each program are a set of real numbers, and we consider two programs to be functionally equal if their unit test output vectors are equal.[1]

Not surprisingly in a class with tens of thousands of participants, the range of the outputs over all of the homework submissions can be quite high even in the simplest programming assignment. Figure 3(a) histograms the 42 assigned problems with respect to the number of distinct unit test outputs submitted by all students. On the low end, we observe that the 32,876 submissions to the simple problem of constructing a 5 × 5 identity matrix resulted in 218 distinct unit test output vectors. In some sense, the students came up with 217 wrong ways to approach the identity matrix problem. The median number of distinct outputs over all 42 problems was 423, but at the high end, we observe that the 39,421 submissions to a regularized logistic regression problem produced 2,992 distinct unit test outputs!

But were there *truly* nearly 3,000 distinct wrong ways to approach regularized logistic regression? Or were there only a handful of "typical" ways to be wrong and a large number of submissions which were each wrong in their own unique way? In the following, we say that a unit test output vector v *owns* a submission

[1] The analysis in Section 4 captures variability of programs at a more nuanced level of detail



(a) (b)

Fig. 4. (a) Number of students covered by the 50 most common unit test outputs for several representative problems; (b) Histogram over the 42 problems of number of students covered by the top 50 unit test outputs for each problem. Observe that for most problems, 50 unit test outcomes is sufficient for covering over 90% of students.

if that submission produced v when run against the given unit tests. We are interested in common or "popular" outputs vectors which own many submissions.

Figure 3(b) visualizes the popularity of the 50 unit class output vectors which owned the most submissions for the gradient descent for linear regression problem. As with all problems, the correct answer was the most popular, and in the case of linear regression, there were 28,605 submissions which passed all unit tests. Furthermore, there were only 15 additional unit test vectors which were the result of 100 submissions or more, giving some support to the idea that we can "cover" a majority of submissions simply by providing feedback based on a handful of the most popular unit test output vectors. On the other hand, if we provide feedback for only a few tens of the most popular unit test outputs, we are still orphaning in some cases thousands of submissions. Figure 3(c) plots the fraction of output vectors for the linear regression problem again which own less than k submissions (varying k on a logarithmic scale). The plot shows, for example, that approximately 95% of unit test output vectors (over $1,000$ in this case) owned 10 or fewer submissions. It would have been highly difficult to provide feedback for this 95% using the vanilla output-based feedback strategy.

To better quantify the efficacy of output-based feedback, we explore the notion of *coverage* — we want to know how many students in a MOOC we can "cover" (or provide output-based feedback for) given a fixed amount of work for the teaching staff. To study this, consider a problem P for which unit test output vectors $S = \{s_1, \ldots, s_k\}$ have been manually annotated by an instructor. This could be as simple as "good job!", to "make sure that your for-loop covers special case X". We say that a student is covered by S if every submitted solution by that student for problem P produces unit test outputs which lie in S. Figure 4(a) plots the number of students which are covered by the 50 most common unit test output vectors for several representative problems. By and large, we find that annotating the top 50 output vectors yields coverage of 90% of students or more in almost all problems (see Figure 4(b) for histogrammed output coverage over the 42 problems). However, we note that in a few cases, the top 50 output vectors might only cover slightly over 80% of students, and that even at

90% coverage, typically between 1000-2000 students are *not* covered, showing limitations of this "vanilla" approach to output-based feedback.

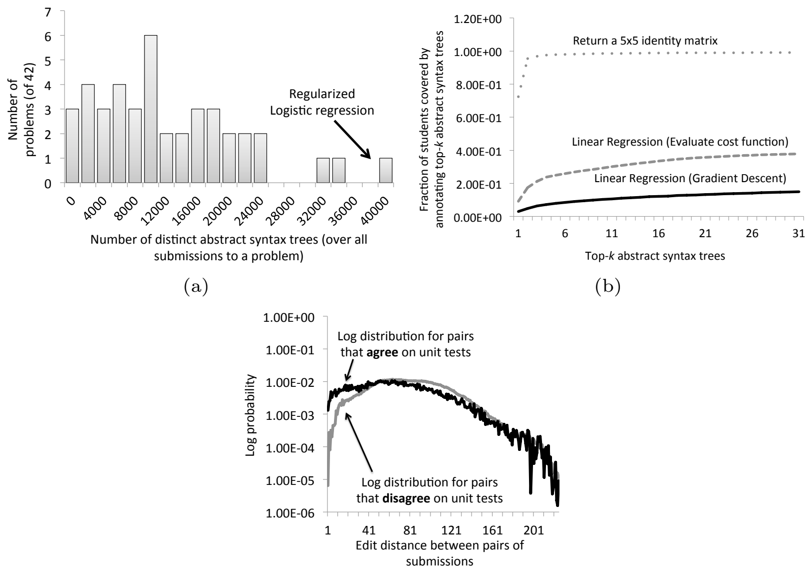
Thus, while output-based feedback provides us with a useful start, the vanilla approach has some limitations. More importantly however, output based feedback can often be too much of an oversimplification. For example, output-based feedback does not capture the fact that multiple output vectors can result from similar misconceptions and conversely that different misconceptions can result in the same unit test outputs. Success of output-based feedback depends greatly on a well designed battery of unit tests. Moreover, coding style which is a critical component of programming cannot be captured at all by unit test based approaches to providing feedback. In the next sections, we discuss a deeper analysis which delves further into program structure and is capable of distinguishing the more stylistic elements of a submission.

4 Syntactic variability of code submissions

In addition to providing feedback on the functional output of a student's program, we also investigate our ability to give feedback on programming style. The syntax of code submission in its raw form is a string of characters. While this representation is compact, it does not emphasize the meaning of the code. To more accurately capture the structure of a programming assignment, we compare the corresponding Abstract Syntax Tree (AST) representation.

This task is far more difficult due to the open ended nature of programming assignments which allows for a large space of programs. There were over half a million unique ASTs in our dataset. Figure 5(b) shows that homework assignments had substantially higher syntactic variability than functional variability. Even if a human labeled the thirty most common syntax trees for the Gradient Descent part of the Linear Regression homework, the teacher annotations would cover under 16% of the students. However, syntactic similarity goes beyond binary labels of "same" or "different". Instead, by calculating the *tree edit distance* between two ASTs we can measure the degree to which two code submissions are similar. Though it is computationally expensive to calculate the similarity between all pairs of solutions in a massive class, the task is feasible given the dynamic programming edit distance algorithm presented by Shasha et al [5] . While the algorithm is quartic in the worst case, it is quadratic in practice for student submission. By exploiting the [5] algorithm and using a computing cluster, we are able to match submissions at MOOC scales.

By examining the network of solutions within a cutoff edit distance of 5, we observe a smaller, more manageable number of common solutions. Figure 1 visualizes this network or landscape of solutions for the linear regression (with gradient descent) problem, with node representing a distinct AST and node sizes scaling logarithmically with respect to the number of submissions owned by that AST. By organizing the space of solutions via this network, we are able to see clusters of submissions that are syntactically similar, and feedback for one AST could potentially be propagated to other ASTs within the same cluster.



(a)

(b)

(c)

Fig. 5. (a) Histogram of the number of distinct abstract syntax trees (ASTs) submitted to each problem.; (b) Number of students covered by the 30 most common ASTs for several representative problems; (c) (Log) distribution over distances between pairs of submissions for pairs who agree on unit test outputs, and pairs who disagree. For very small edit distances (<10 edits), we see that the corresponding submissions are typically also functionally similar (i.e., agree on unit test outputs).

Figure 1 also encodes the unit test outputs for each node using colors to distinguish between distinct unit test outcomes.[2] Note that visually, submissions belonging to the same cluster typically also behave similarly in a functional sense, but not always. We quantify this interaction between functional and syntactic similarity in Figure 5(c) which visualizes (log) distributions over edit distances between pairs of submissions who *agree* on unit test outcomes and pairs of submissions who *disagree* on unit test outcomes. Figure 5(c) shows that when two ASTs are within approximately 10 edits from each other, there is a high probability that they are also functionally similar. Beyond this point, the two distributions are not significantly different, bearing witness to the fact that programs that behave similarly can be implemented in significantly different ways.

5 Discussion and ongoing work

The feedback algorithm outlined in this paper lightly touches on the potential for finding patterns that can be utilized to force multiply teacher feedback. One

[2] Edge colors are set to be the average color of the two endpoints.

clear path forward is to propagate feedback, not just for entire programs, but also for program parts. If two programs are different yet share a substantial portion in common we should be able to leverage that partial similarity.

Though we focused our research on creating an algorithm to semi-automate teacher feedback in a MOOC environment, learning the underlying organization of assignment solutions for an entire class has benefits that go beyond those initial objectives. Knowing the space of solutions and how students are distributed over that space is valuable to teaching staff who could benefit from a more nuanced understanding of the state of their class. Moreover, though this study is framed in the context of MOOCs, the ability to find patterns in student submissions should be applicable to any class with a large enough corpus of student solutions, for example, brick and mortar classes which give the same homeworks over multiple offerings, or Advanced Placement exams where thousands of students answer the same problem.

References

1. D. Gitchell and N. Tran. Sim: a utility for detecting similarity in computer programs. In *ACM SIGCSE Bulletin*, volume 31, pages 266–270. ACM, 1999.
2. B. Hartmann, D. MacDougall, J. Brandt, and S. R. Klemmer. What would other programmers do: suggesting solutions to error messages. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 1019–1028. ACM, 2010.
3. C. Piech, M. Sahami, D. Koller, S. Cooper, and P. Blikstein. Modeling how students learn to program. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, pages 153–160. ACM, 2012.
4. P. M. Sadler and E. Good. The impact of self-and peer-grading on student learning. *Educational assessment*, 11(1):1–31, 2006.
5. D. Shasha, J. T.-L. Wang, K. Zhang, and F. Y. Shih. Exact and approximate algorithms for unordered tree matching. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(4):668–678, 1994.

Revisiting and Extending the Item Difficulty Effect Model

Sarah Schultz and Trenton Tabor
Worcester Polytechnic Institute
100 Institute Rd, Worcester, MA
{seschultz, tstabor}@wpi.edu

Abstract: Data collected by learning environments and online courses contains many potentially useful features, but traditionally many of these are ignored when modeling students. One feature that could use further examination is item difficulty. In their KT-IDEM model, Pardos and Heffernan proposed the use of question templates to differentiate guess and slip rates in knowledge tracing based on the difficulty of the template- here, we examine extensions and variations of that model. We propose two new models that differentiate based on template- one in which the learn rate is differentiated and another in which learn, guess, and slip parameters all depend on template. We compare these two new models to knowledge tracing and KT-IDEM. We also propose a generalization of IDEM in which, rather than individual templates, we differentiate between multiple choice and short answer questions and compare this model to traditional knowledge tracing and IDEM. We test these models using data from ASSISTments, an open online learning environment used in many middle and high school classrooms throughout the United States.

Keywords: Knowledge tracing, student modeling, item difficulty, Bayesian networks, educational data mining

1. Introduction

Traditionally, knowledge tracing (KT), does not take into account much of the data collected by tutoring system. Some work has been done on leveraging hint and attempt counts in KT [8], [9], and in individualizing based on student [6], but one area that merits more exploration is the use of item difficulty to more accurately model students. Pardos and Heffernan proposed a model to do just that [5], but explored only one such possible model. We created two variations on this model and a generalization of it in order to determine which of these models is the best predictor of student knowledge. Our goal is to discover how item difficulty really affects students' knowledge and performance.

2. Models

2.1 Knowledge Tracing

In classic knowledge tracing [1], the goal is to predict whether a student will answer the next question correctly based upon the current estimate of their knowledge. In the Bayesian network, the responses are the observed nodes, and the student's knowledge at each time-step are the latent nodes. Using Expectation Maximization (EM) or another algorithm, we learn values for the probability of initial knowledge, P(L_0); the probability of learning the skill from one time step to the next, P(T); the probability of guessing correctly when the skill is in the unlearned state, P(G); and the probability of slipping, or answering incorrectly when the skill is in the learned state, P(S) (Figure 1).

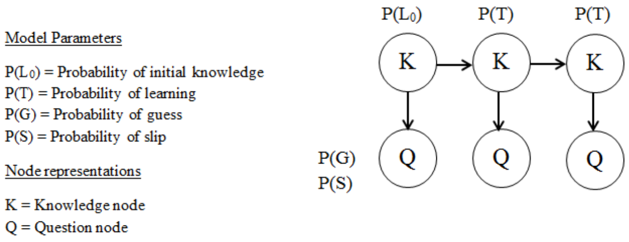

Model Parameters
P(L0) = Probability of initial knowledge
P(T) = Probability of learning
P(G) = Probability of guess
P(S) = Probability of slip
Node representations
K = Knowledge node
Q = Question node
P(L0)
P(T)
P(T)
K
K
K
Q
Q
Q
P(G)
P(S)

Fig. 1- Standard Knowledge Tracing

2.2 KT-IDEM

In 2011, Pardos and Heffernan proposed the Knowledge Tracing- Item Difficulty Effect Model (KT-IDEM), which adds difficulty to the traditional KT model by adding an item difficulty node affecting the question node. This model learns a separate guess and slip rate for each item, and therefore has N*2+2 parameters, where N is the number of unique items, in comparison to KT's four [5]. Figure 2 illustrates the KT-IDEM model.

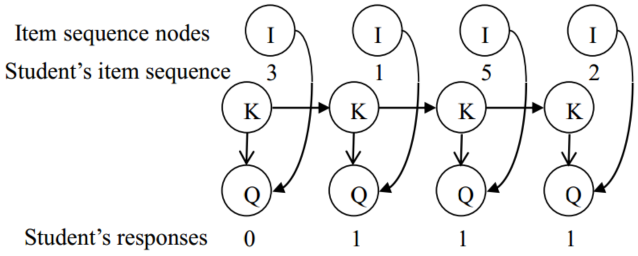


Fig. 2- Knowledge Tracing- Item Difficulty Effect Model

2.3 Extensions to IDEM

We believe that question difficulty not only affects performance, but will also have an effect on learning. By answering questions of different difficulties and receiving feedback on whether or not the answer is correct, students could learn differing amounts. We therefore propose two new variations on KT-IDEM. The first individualizes learn rates by item difficulty, but keeps guess and slip consistent. The second individualizes learn, guess, and slip rates based on item difficulty. In a ten item dataset, KT would have four parameters, KT-IDEM would have 22, the first of our models, Item Difficulty Effect on Learning (IDEL), would have 12, and the second, Item Difficulty Effect All (IDEA), would have 32. It is possible that certain datasets will be over-parameterized in some of these models if there are not enough data points per item, but as Pardos and Heffernan pointed out in their original KT-IDEM paper, “there has been a trend of evidence that suggests models that have equal or even more parameters than data points can still be effective” [5]. These models are illustrated below (Figures 3 and 4).

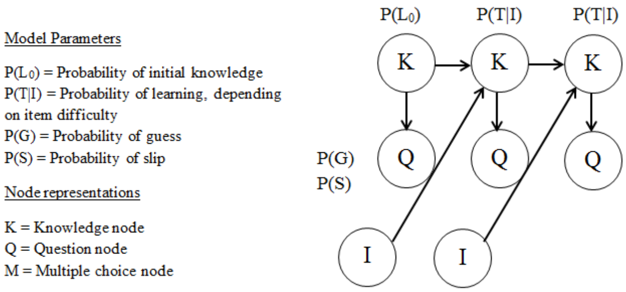

Model Parameters
P(L0) = Probability of initial knowledge
P(T|I) = Probability of learning, depending on item difficulty
P(G) = Probability of guess
P(S) = Probability of slip
Node representations
K = Knowledge node
Q = Question node
M = Multiple choice node
P(L0)
P(T|I)
P(T|I)
K
K
K
P(G)
P(S)
Q
Q
Q
I
I

Fig. 3- Item Difficulty Effect on Learning

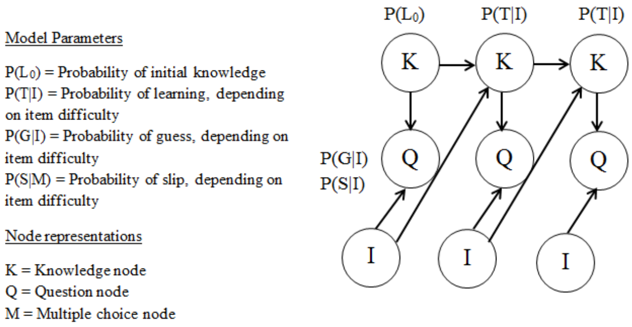

Model Parameters
P(L0) = Probability of initial knowledge
P(T|I) = Probability of learning, depending on item difficulty
P(G|I) = Probability of guess, depending on item difficulty
P(S|M) = Probability of slip, depending on item difficulty
Node representations
K = Knowledge node
Q = Question node
M = Multiple choice node
P(L0)
P(T|I)
P(T|I)
K
K
K
P(G|I)
P(S|I)
Q
Q
Q
I
I
I

Fig. 4- Item Difficulty Effect All

2.4 MC

The final model we implemented is a generalization of KT-IDEM, which adds a multiple choice node to KT at each time step, indicating whether the particular question is multiple choice or not, rather than an item difficulty node. We now learn two different guess and slip rates, one each for multiple choice questions and for non-multiple choice questions. As is standard in KT and all other models explored in this paper, we assume that students do not forget. The multiple choice model (MCKT) is illustrated in Figure 5.

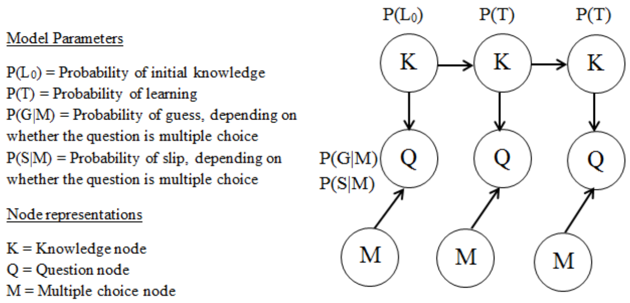

Model Parameters
P(L0) = Probability of initial knowledge
P(T) = Probability of learning
P(G|M) = Probability of guess, depending on whether the question is multiple choice
P(S|M) = Probability of slip, depending on whether the question is multiple choice
Node representations
K = Knowledge node
Q = Question node
M = Multiple choice node
P(L0)
P(T)
P(T)
K
K
K
P(G|M)
P(S|M)
Q
Q
Q
M
M
M

Fig. 5- Multiple Choice Model

We expected that the guess rate for multiple choice questions would be higher than the guess rate for non-multiple choice questions, since there are a finite number of options presented as opposed to an open response where it is possible to enter almost anything. We also expected that the slip rate would be lower for multiple choice questions, as recognizing the correct answer is generally easier than recalling it [3].

3. Dataset

3.1 The ASSISTments Tutoring System

The data used in this work is from ASSISTments, a freely available online mathematics tutoring system for grades 4 to 10 [2]. This system is used in classrooms across the country, and while it is not currently in itself a course, it is certainly an open, large-scale online learning tool.

In ASSISTments, multiple items can be built using the same template, where the only difference is the actual numbers in the problem. We consider problems generated from the same template to be the same item when working with the models that consider item difficulty.

We used six skills from the dataset, all of which came from skill builder data. In ASSISTments, skill builders are sessions where a student practices a certain skill until s/he gets three questions correct in a row, at which point it is considered to be learned. Within each skill, there are different sequences of templates that a student could encounter. In order to be sure that all students in our dataset were seeing the same templates, we used one sequence from each skill, except for Ordering Integers, from which we sampled two sequences separately. Table 1 shows information about the sequences we used in our experiments.

Table 1- Sequences used to test the models

Skill Name	Percent correct	Number of Templates	Percent Multiple Choice
Pythagorean Theorem	34	8	70
Ordering Integers (1)	88	3	34
Ordering Integers (2)	84	3	65
Square Root	89	2	38
Ordering Positive Decimals	74	3	100
Percent	33	13	67
Pattern Finding	48	5	45

4. Methods

Using Kevin Murphy's Bayes Net Toolbox for Matlab [4], we built each of our proposed models. We performed a 5-fold cross-validation on each of the seven sequences from the ASSISTments dataset using all five models, where four folds were used for training and the fifth for testing. The data was partitioned into folds randomly such that each student within a skill was in only one fold and the same folds were used for every model to guarantee a fair comparison. To avoid over-fitting the models to any student who practiced a skill a large number of times, only the first five opportunities of the skill for each student were used. We used expectation maximization to learn the parameters for each of our models.

5. Results

In order to compare models, we calculated mean absolute error (MAE), root mean square error (RMSE), and area under the curve (AUC) of each model's predictions compared to the actual data. We performed a paired t-test of each of these measures using the runs from each fold and found that RMSE was the most consistently reliable measure, so we use that to determine which model is best. Table 2 shows an example of all metrics, obtained from the skill "Percent," which has 13 templates. From this data, it appears that KT has the worst MAE and AUC of all the models, but KT-IDEL has a worse RMSE.

Table 2- Results for “Percent”

	Knowledge Tracing	KT-IDEM	KT-IDEL	KT-IDEA	MCKT
MAE	0.433231	0.350409	0.433039	0.352525	0.352107
AUC	0.531074	0.762205	0.56607	0.706951	0.754057
RMSE	0.472552	0.449915	0.481702	0.441461	0.462738

Comparing the template-based models to KT, we found that for this skill, the MAE was reliably better for KT-IDEM than KT or KT-IDEL and the AUC of KT-IDEM was reliably better than KT and both other template models. On the other hand, KT-IDEA had a reliably better RMSE than KT-IDEM for this skill.

Taking the data from all seven sequences, we unfortunately did not find a conclusive answer to the question of which template-based model performs best. For the skill “Pattern Finding,” we found that KT-IDEM did best in all three measures, whereas for the first sequence of “Ordering Integers,” KT-IDEL outperformed the other two template-based models, but was not significantly different from KT. (A few additional results tables can be found in the appendix of this paper.)

Our next question, was whether the multiple choice model would perform better than KT or KT-IDEM. While theoretically, the multiple choice model should be the same as KT when all problems are of one type, when we ran the models over a sequence that was all multiple choice, the models learned different parameters. This is probably because the multiple choice nodes must always have two values in their CPT tables. We therefore exclude this sequence from analysis of the multiple choice model. On the other hand, we did test a sequence that was all one template, and all template models behaved the same, since the number of values in the template nodes’ CPT tables is the same as the number of templates. Out of the six remaining sequences in which we can compare MCKT, each with three metrics, for a total of 18 comparisons, we found that MCKT was reliably better than KT six times, and reliably better than KT-IDEM four times. Out of these, only two instances showed MCKT better than both of the other models. Out of the remaining nine comparisons, four showed that MCKT was better than the others, but not reliably so, in one case KT-IDEM outperforms MCKT, which is marginally better than KT, and in six cases the both of the other models performed better than MCKT. Since MCKT is at least marginally better than KT a majority of the time, and significantly better in 6 out of 18 cases, it looks like it could be a promising model, although more research is needed.

6. Contributions and Future Work

In this work, we proposed three new models; IDEL, IDEA, and MCKT. We compared these models to traditional KT and to KT-IDEM and found that different models worked best for different sequences. Our findings are not in agreement with [5], which states that IDEM works better than KT in ASSISTments skill builder data, and our observations also seem to indicate that other item difficulty models could work better than KT-IDEM. The interesting contribution here is that this means question difficulty does, in fact, appear to affect learning, possibly more than performance on the current item.

We used only six sequences (and had to exclude one from analysis), all from the same system, in this preliminary look at these models and would like to, in the future, try using more sequences and data from other tutors to see be sure that findings hold true in other scenarios and are not useful only in ASSISTments. Although, even if the latter is the case, having a better student modeling technique for this system would be very useful in developing ways to make it better.

One clear next step is to implement the same extensions made to the IDEM model to the multiple choice model in order to determine how the different types of questions- multiple choice and short answer- effect student knowledge and performance.

Acknowledgements

This work was supported by the WPI Computer Science department and Robotics Engineering program. We would like to thank all of the people associated with creating ASSISTments, as well as the schools and teachers using ASSISTments. We thank Professor Neil Heffernan for helping us to conceptualize this work and for his ideas and Yutao Wang for the code base for knowledge tracing. We would also like to thank Zach Pardos for his advice.

Appendix

Table 3- Results for “Pythagorean Theorem”

	KT	KT-IDEM	KT-IDEL	KT-IDEA	MCKT
MAE	0.480245	0.448852	0.478075	0.431558	0.472431
AUC	0.610767	0.630755	0.661355	0.671785	0.587751
RMSE	0.491635	0.517432	0.487239	0.511354	0.530694

Table 4- Results for “Ordering Positive Decimals” (MCKT excluded)

	KT	KT-IDEM	KT-IDEL	KT-IDEA
MAE	0.352754	0.434477	0.362968	0.451735
AUC	0.58984	0.549476	0.61913	0.577328
RMSE	0.422419	0.474215	0.418596	0.492843

Table 5- Results for “Ordering Positive Integers (1)”

	KT	KT-IDEM	KT-IDEL	KT-IDEA	MCKT
MAE	0.223823	0.268527	0.223668	0.270949	0.2948
AUC	0.545965	0.351229	0.560837	0.36537	0.38731
RMSE	0.333427	0.365692	0.335122	0.394251	0.3903

References

1. Corbett, A.T. and Anderson, J.R., 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modeling and User Adapted Interaction, 4(4), pp.253–278.
2. Heffernan, N.T. ASSISTments. http://teacherwiki.assistment.org/wiki/About www.assistments.org
3. Moreno, R., 2010. Education Psychology, John Wiley & Sons, Inc.
4. Murphy, K. 2007. Bayes Net Toolbox for Matlab.
5. Pardos, Z.A. and Heffernan, N.T., 2011. KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. In J. A. Konstan, R. Conejo, J. L. Marzo, & N. Oliver, eds. User Modeling, Adaption and Personalization. Springer Berlin Heidelberg, pp. 243–254.
6. Pardos, Z.A. and Heffernan, N.T., 2010. Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. User Modeling Adaptation and Personalization, In press, pp.255–266.
7. Qiu, Y., Qi, Y., Lu, H., Pardos, Z. and Heffernan, N., 2011. Does time matter modeling the effect of time in Bayesian knowledge tracing. In M. Pechenizkiy, T. Calders, C. Conati, S. Ventura, C. Romero, & J. Stamper, eds. Proceedings of the 4th International Conference on Educational Data Mining. pp. 139–148.
8. Wang, Y. and Heffernan, N.T., 2010. Leveraging First Response Time into the Knowledge Tracing Model.
9. Wang, Y. and Heffernan, N.T., 2011. The “Assistance” model: Leveraging how many hints and attempts a student needs. In 24th International Florida Artificial Intelligence Research Society FLAIRS 24 May 18 2011 May 20 2011. AAAI Press, pp. 549–554.

Using Argument Diagramming to Improve Peer Grading of Writing Assignments

Mohammad H. Falakmasir[1], Kevin D. Ashley
Christian D. Schunn

Intelligent Systems Program, Learning Research and Development Center,
University of Pittsburgh
{mhf11, ashley, schunn}@pitt.edu

Abstract. One of the major components of MOOCs is the weekly assignment. Most of the assignments are multiple choice, short answer or programming assignments and can be graded automatically by the system. Since assignments that include argumentation or scientific writing are difficult to grade automatically, MOOCs often use a crowd-sourced evaluation of the writing assignments in the form of peer grading. Studies show that this peer-grading scheme faces some reliability issues due to widespread variation in the course participants' motivation and preparation. In this paper we present a process of computer-supported argumentation diagramming and essay writing that facilitates the peer grading of the writing assignments. The process has not been implemented in a MOOC context but all the supporting tools are web-based and can be easily applied to MOOC settings.

Keywords: Computer Supported Argumentation, Argument Diagramming, Peer Review and Grading

1 Introduction

MOOCs in general and Coursera, in particular, started with courses in the area of Computer Science. These courses offered a variety of homework including multiple choice, short answer, and programming assignments that can be graded automatically by the system. However, recently, many MOOCs have started offering courses in social sciences, humanities, and law subjects whose assignments naturally involve more writing and argumentation. Automatic grading of those kinds of assignments is more challenging given the current state of natural language processing technologies. Coursera and most of the other current systems use a peer-grading mechanism in order to address this issue. However, because of the open access nature of the MOOCs, a massive number of people with different educational backgrounds and language skills from all around the world participate in these courses and this heterogeneity in prior preparation negatively affects the validity and reliability of

[1] Corresponding Author

peer-grades. Researchers have investigated this issue (Duneier, 2012) and some steps have been taken to address it. Coursera, for example, flags students who give inaccurate grades and assigns their assessments less weight, but this method does not directly address the diversity of knowledge and writing skills among the students. In this paper, we recommend an approach to this issue that combines computer-supported argument diagramming and writing with scaffolded peer-review and grading. With support of the National Science Foundation,[2] our ArgumentPeer process combines two web-based tools (SWoRD and LASAD) that have been used in several university settings and courses, and applies them to support argumentation and writing assignments in science and law. The process enables the instructional team to carefully define and monitor the writing assignment and revision procedure and involves several machine learning and natural language processing components.

2 Background

Writing and argumentation are fundamental skills that support learning in many topics. Being able to understand the relationships among abstract ideas, to apply them in solving concrete problems, and to articulate the implications of different findings for studies and theories are essential for students in all areas of science, engineering, and social studies. However, inculcating these skills, or compensating for the lack of them, is especially difficult in MOOC setting where students have such diverse preparations and motivations.

Our approach to tackle this problem involves breaking down the process of writing into multiple measurable steps and guiding the student through the steps with careful support and feedback. The first step of the process, computer-supported argument planning, engages the students with a graphical representation for constructing arguments and provides them with feedback and intelligent support. We use LASAD[3] as our argument-diagramming tool (cf. Scheuer et al., 2010). LASAD is a web-based argumentation support system to help students learn argumentation in different domains. It supports flexible argument diagramming by enabling instructors to define a pre-structured palette of argumentation elements (Argument Ontology) along with a set of help system rules in order to give instant feedback to students while working on their diagrams.

The massive number of students in MOOC settings makes it impossible for the instructional team to provide reflective feedback on each individual student's argument. We handle this issue with computer-supported peer-review and grading using SWoRD[4] (Cho & Schunn, 2007). In general, peer review is consistent with learning theories that promote active learning. Furthermore, the peer-review of writing has some learning benefits for the reviewer, especially when the students provide constructive feedback (Wooley, Was, Schunn, & Dalton, 2008), and put effort into the process (Cho & Schunn, 2010). Moreover, studies have shown that

[2] Award No.1122504 DIP: Teaching Writing and Argumentation with AI-Supported Diagramming and Peer Review

[3] http://cscwlab.in.tu-clausthal.de/lasad/

[4] https://sites.google.com/site/swordlrdc/

feedback from a group of peers can be at least as useful as that of teachers (Cho & Schunn, 2007), especially when good rubrics and incentives for reviewing are included. Most relevant here, studies have shown that even students with lower levels of knowledge in the topic can provide feedback that is useful to the ones with higher levels (Patchan & Schunn, 2010; Patchan, 2011).

3 The Process

The ArgumentPeer process includes two main phases: 1) Argument Planning, and 2) Argument Writing. Fig. 1 shows an overview of the process and its underlying components and steps.

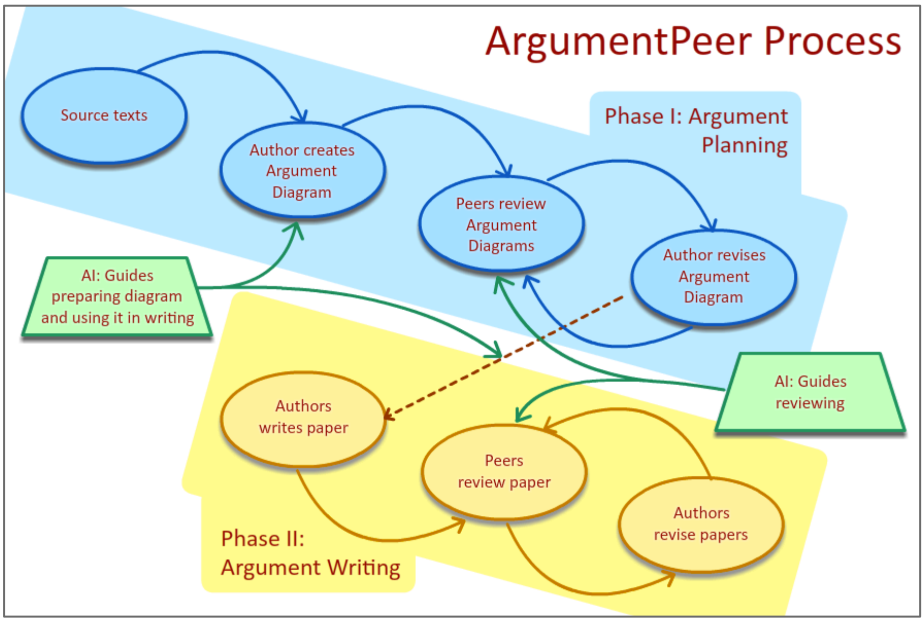

ArgumentPeer Process
Phase I: Argument Planning
Source texts
Author creates Argument Diagram
Peers review Argument Diagrams
Author revises Argument Diagram
AI: Guides preparing diagram and using it in writing
AI: Guides reviewing
Authors writes paper
Peers review paper
Authors revise papers
Phase II: Argument Writing

Fig. 1: ArgumentPeer Process

3.1 Phase I: Argument Diagramming

This phase includes studying the assigned resources and creating the argument diagram. As an example, students in a legal writing course used LASAD in order to prepare textual brief on appeal to the U.S. Supreme Court in the case of *United States v. Alvarez* (Lynch et al., 2012). The system had been introduced to them in a 45-minutes lecture session (that could easily be made a video) and students were directed toward a recommended stepwise format for written legal argumentation as set forth in a noted authority (Neumann 2005). Figure 2 shows an example diagram in this study.

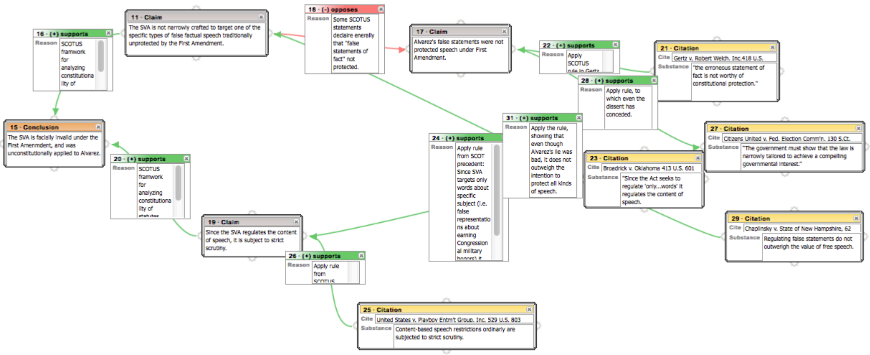

16 · (+) supports
Reason
SCOTUS framwork for analyzing constitutionality of
11 · Claim
The SVA is not narrowly crafted to target one of the specific types of false factual speech traditionally unprotected by the First Amendment.
18 · (-) opposes
Reason
Some SCOTUS statements declaire enerally that "false statements of fact" not protected.
17 · Claim
Alvarez's false statements were not protected speech under First Amendment.
22 · (+) supports
Reason
Apply SCOTUS rule in Gertz
21 · Citation
Cite
Gertz v. Robert Welch. Inc.418 U.S.
Substance
"the erroneous statement of fact is not worthy of constitutional protection."
28 · (+) supports
Reason
Apply rule, to which even the dissent has conceded.
15 · Conclusion
The SVA is facially invalid under the First Amenmdent, and was unconstitutionally applied to Alvarez.
20 · (+) supports
Reason
SCOTUS framwork for analyzing constitutionality of statutes
24 · (+) supports
Reason
Apply rule from SCOT precedent: Since SVA targets only words about specific subject (i.e. false representations about earning Congressional military honors) it
31 · (+) supports
Reason
Apply the rule, showing that even though Alvarez's lie was bad, it does not outweigh the intention to protect all kinds of speech.
23 · Citation
Cite
Broadrick v. Oklahoma 413 U.S. 601
Substance
"Since the Act seeks to regulate 'only...words' it regulates the content of speech.
27 · Citation
Cite
Citizens United v. Fed. Election Comm'n. 130 S.Ct.
Substance
"The government must show that the law is narrowly tailored to achieve a compelling governmental interest."
19 · Claim
Since the SVA regulates the content of speech, it is subject to strict scrutiny.
26 · (+) supports
Reason
Apply rule from SCOTUS
29 · Citation
Cite
Chaplinsky v. State of New Hampshire, 62
Substance
Regulating false statements do not outwerigh the value of free speech.
25 · Citation
Cite
United States v. Playboy Entm't Group. Inc. 529 U.S. 803
Substance
Content-based speech restrictions ordinarly are subjected to strict scrutiny.

Fig. 2: Example Argument Diagram in Legal Writing Course

The instructional team tailored the argument ontology to support the recommended argumentation format; the nodes were basically legal "claim" and "conclusion" nodes that are connected together via "supporting" and "opposing" links providing reasons for and against. The development of a suitable ontology is a critical aspect in the design of an argumentation system and might involve iterative refinement based on observed problems and weaknesses (Buckingham et al., 2002). Specifically, ontologies affect the style of argumentation (Suthers, et al., 2001) and the level of details expected for students to provide. LASAD provides an authoring tool that enables the instructional team to carefully design the argumentation ontology.

After creating the argument diagrams, the students submit their diagrams to the SWoRD system for revision. As noted, SWoRD lets instructors provide a detailed rubric with which peers should assess the diagram. Moreover, it has a natural language processing (NLP) component that pushes reviewers to provide useful feedback that is not ambiguous or vague (more details in section 3.3). After receiving the reviews, the author will revise his/her argument diagram and get ready to write the first draft of the writing assignment in phase 2. To support this transition to a written argument, a system component creates a textual outline based on a depth-first traversal of the argumentation diagram and informed by the argument ontology. In this way, students are encouraged to create a well-annotated argumentation diagram because the diagram text is easily transferred directly to the written draft.

3.2 Phase II: Writing

In this phase, students write their first drafts using the outlines generated from the argument diagrams and submit them to SWoRD. After that, the system automatically assigns the draft to *n* reviewers based on the instructors' policy. The instructor can also assign the individual or groups of peers for the revision using various methods. For example, in the Legal Writing course, the instructor divided the students into two groups, one, writing for the majority and the other writing for the dissenting judge in the 9th Circuit U.S. Court of Appeals and assigned the peers in a way such that there is at least one peer from the other group among the reviewers.

In the next step, the instructor carefully designs the paper reviewing criteria (rubric) for the peers and then starts the reviewing process. The key feature of SWoRD is the ease with which instructors can define rubrics to guide peer reviewers in rating and commenting upon authors' work. The instructor-provided rubrics, which may include both general domain writing and content-specific criteria (Goldin & Ashley, 2012), should help to focus peer feedback and compensate for the wide diversity of peer-reviewers' preparation and motivation.

Reviewers, then, download the paper and evaluate them based on the defined rubric and submit their reviews and ratings to SWoRD. Again, the NLP component of the system, checks the reviews for usefulness and then the system deliverers the reviews back to the author. SWoRD automatically determines the accuracy of each reviewer's numerical ratings using a measure of consistency applied across all of the writing dimensions (Cho & Schunn, 2007). Finally, the author submits the second draft to the system and the final draft can either be grader by peers or the instructional team, although of course in a MOOC context peers would grade it again.

3.3 AI Guides Student Authors and Reviewers in Both Phases

As mentioned, the LASAD Authoring tool and its flexible ontology structure enable instructors to specify the level of detail on which they want the students to focus. Instructors can also use the Feedback Authoring tool to define help system rules that guide the students through the argumentation diagramming process. The instant feedback component of LASAD is an expert system that uses logical rules to analyze students' developing argument diagrams and to provide feedback on making more complete and correct diagrams. The hints can be as simple as telling the student to fill in a text field for an element, or as complex as telling the student to include opposing, as well as supporting, citations for a finding. Using this in-depth intervention, instructors can focus students on their intended pedagogical goals. For example, in the legal writing course, a help system rule asks students to include at least one opposing "citation" in their diagrams to anticipate possible important counterarguments that a court would expect an advocate to have addressed in his or her brief.

The NLP component of SWoRD helps the students improve their reviews by detecting the presence or absence of key feedback features like the location of the problem and the presence of an explicit solution. This feature has been implemented for review comments on both argumentation diagrams and the written drafts. The details of the computational linguistic algorithm that detects the feedback issues are described in (Xiong et al., 2012; Nguyen & Litman, in press). The interface provides reviewers with advice like: "Say where this issue happened." "Make sure that for every comment below, you explain where in the paper it applies." In addition, it provides examples of the kind of good feedback likely to result in an effective revision: "For example, on page [x] paragraph [y], …. Suggest how to fix this problem." "For example, when you talk about [x], you can go into more detail using quotes from the reading resource [y]." The system tries to be as helpful as possible, but in order to prevent frustration, it allows the reviewers to ignore the suggestions and submit the review as is. However, SWoRD considers these reviewers as less accurate and gives lower weight to their ratings.

4 Assessment and Grading

After submitting the final draft, the papers are assigned automatically or by the instructors to the same or another group of peers (or members of the instructional team in non-MOOC contexts) for grading. The same rubric can be used for the second round of review but it is also possible to define new criteria particularly for grading purposes.

According to (Cho, Schunn, & Wilson, 2006; Patchan, Charney, & Schunn, 2009) the aggregate ratings of at least 4 peers on a piece of writing in this setting are more highly reliable and just as valid as a single instructor's ratings. However, some studies (e.g., Chang et al., 2011) note that there can be systematic differences between peer and instructor assessment in a web-based portfolio setting. We believe that by breaking down the argument planning and writing process into multiple guided steps, each subject to review according to instructor-designed peer-review criteria, we move toward a more reliable peer-grading scheme that can be especially useful in a MOOC context.

5 Discussion

Grading writing assignments requires considerable effort, especially when the class size increases. Peer-review and grading is one way to deal with this problem but many instructors are hesitant to use it in their classrooms. The main concern is whether the students are actually capable of grading the papers accurately and responsively. Studies have shown that peer rating alone can be reliable and valid in a large-scale classroom under appropriate circumstances and well-chosen review criteria (Cho, Schunn, & Wilson, 2006; Patchan, Charney, & Schunn, 2009). The ArgumentPeer project not only enables the instructor to design the rubric but also makes it salient for the reviewer to see the deep structure of the argumentation by viewing the argumentation diagram. This positive synergy between diagramming and peer-review makes it easier for the reviewer to see the argument structure in the diagram and its reflection in the writing.

Regarding scalability and the possibility of being used in a MOOC setting, both SWoRD and LASAD are web-based projects developed using Java 2 Platform, Enterprise Edition (J2EE) architecture. LASAD uses automated load balancing in order to support a large number of students. The rich graphical interface of LASAD along with flexible structure of the ontologies helps students gain an understanding of the topic of argumentation (Loll, et al., 2010). Moreover, the collaborative nature of LASAD can be used in order to facilitate engagement, particularly in MOOC settings that face the problem of student retention.

SWoRD, which is the main platform for peer-review and grading, has also been successfully used in classrooms with a large number of students (Cho, Schunn, & Wilson, 2006). The basic review structure in SWoRD is quite similar to the journal publication process, which makes it a familiar process among academics. In addition, publicizing students' papers to their peers can make students put more effort into writing by increasing audience awareness (Cohen & Riel, 1989).

6 Conclusion

In this paper, we presented a process of argument diagramming and reciprocal peer-review in order to facilitate the grading of writing assignments. The ArgumentPeer process and its preexisting components, SWoRD and LASAD, have been applied across different university settings in different courses with large numbers of students. We have decomposed writing assignments into separate steps of planning an argument and then writing it, support students in each step with instructor- and AI-guided peer reviewing and grading. The results of our past studies show that high reliability and validity in the peer grading can be achieved with multiple reviewers per paper. The web-based nature of the components of the ArgumentPeer process makes it relatively easy to apply in MOOC settings. We believe that its fine-grained support for authoring and reviewing could help achieve higher levels of reliability and validity in MOOCs despite their massive numbers of highly diverse participants.

References

1. Buckingham Shum, S. J., Uren, V., Li, G., Domingue, J., Motta, E., & Mancini, C. (2002). Designing representational coherence into an infrastructure for collective sense-making. Invited discussion paper presented at the 2nd International Workshop on Infrastructures for Distributed Collective Practices.
2. Chang, C. C., Tseng, K. H., & Lou, S. J. (2011). A comparative analysis of the consistency and difference among teacher-assessment, student self-assessment and peer-assessment in a Web-based portfolio assessment environment for high school students. *Computers and Education*, 58(1), 303-320.
3. Cho, K., & Schunn, C. D. (2007). Scaffolded writing and rewriting in the discipline: A web-based reciprocal peer review system. *Computers & Education*, 48(3), 409–426.
4. Cho, K., & Schunn, C. D. (2010). Developing writing skills through students giving instructional explanations. In M. K. Stein & L. Kucan (Eds.), *Instructional Explanations in the Disciplines: Talk, Texts and Technology*. New York: Springer.
5. Cho, K., Schunn, C. D., & Wilson, R. W. (2006). Validity and reliability of scaffolded peer assessment of writing from instructor and student perspectives. *Journal of Educational Psychology*, 98(4), 891-901.
6. Cohen, M., & Riel, M. (1989). The effect of distant audiences on students' writing. *American Educational Research Journal*, 26, 143–159.
7. Duneier, M. (2012). Teaching to the world from central New Jersey. *Chronicle of Higher Education*, September 3.
8. Goldin, I. M. & Ashley, K. D. (2012) Eliciting Formative Assessment in Peer Review. *Journal of Writing Research* 4(2) pp. 203–237.
9. Loll, F., Scheuer, O., McLaren, B. M. & Pinkwart, N. (2010). Computer-Supported Argumentation Learning: A Survey of Teachers, Researchers, and System Developers. In M. Wolpers, P. A. Kirschner, M. Scheffel, S. Lindstaedt, & V. Dimitrova, Proceedings of the 5th European Conference on Technology Enhanced Learning (EC-TEL 2010), LNCS 6383, pp. 530-535. Springer.
10. Lynch, C., Ashley, K. D., Falakmassir, M. H., Comparing Argument Diagrams, in proceedings of The 25th Annual Conference on Legal Knowledge and Information Systems (JURIX), Amsterdam, Netherlands, December 2012, pp. 81-90.

11. Neumann, R. (2005) *Legal Reasoning and Legal Writing: Structure, Strategy, and Style*. (5th Ed.) Walters Kluwer.
12. Nguyen H., Litman D., (in press). Identifying Localization in Peer Reviews of Argument Diagrams. Accepted in the 16th International Conference on Artificial Intelligence in Education (AIED 2013), Memphis, TN.
13. Patchan, M. M., Charney, D., & Schunn, C. D. (2009). A validation study of students' end comments: Comparing comments by students, a writing instructor, and a content instructor. *Journal of Writing Research*, 1(2), 124-152.
14. Patchan, M. M., & Schunn, C. D. (2010). Impact of Diverse Abilities on Learning to Write through Peer-Review. Paper presented at the 32nd annual meeting of the Cognitive Science Society, Portland, OR.
15. Scheuer, O., Loll, F., Pinkwart, N. and McLaren, B. M. (2010). Computer-supported argumentation: A review of the state-ofthe-art. *International Journal on Computer Supported Collaborative Learning*, 5(1), 43-102. Springer.
16. Suthers, D. D., Connelly, J., Lesgold, A., Paolucci, M., Toth, E. E., Toth, J., & Weiner, A. (2001). Representational and advisory guidance for students learning scientific inquiry. In K. D. Forbus & P. J. Feltovich (Eds.), *Smart machines in education: The coming revolution in educational technology* (pp. 7–35). Menlo Park, CA: AAAI/MIT Press.
17. Wooley, R., Was, C., Schunn, C., & Dalton, D. (2008). The effects of feedback elaboration on the giver of feedback. Paper presented at the 30th Annual Meeting of the Cognitive Science Society.
18. Xiong, W., Litman, D., & Schunn, C. D. (2010). Natural Language Processing techniques for researching and improving peer feedback. *Journal of Writing Research*, 4(2), 155-176.

Improving learning in MOOCs with Cognitive Science

Joseph Jay Williams[1]
[1] University of California at Berkeley
joseph_williams@berkeley.edu

Abstract. MOOCs and other platforms for online education are having a tremendous impact on the learning of tens of thousands of students. They offer a chance to build a set of educational resources from the ground up, at a time when scientists know far more about learning and teaching than at the advent of the current education system. This paper presents practical implications of research from cognitive science, showing empirically supported and actionable strategies any designer or instructor can use to improve students' learning. These all take the form of augmenting online videos and exercises with questions and prompts for students to consider explanations: *before, during,* and *after* learning. This class of instructional strategies provides students with direction while allowing them to take charge of their learning, is technically easy to implement, and is applicable to a wide variety of video and exercise content, that ranges across multiple topics.

Keywords: learning, learning, cognitive science, MOOCs, educational software, online learning, problem based learning, explanation, self-explanation, retrieval practice, interleaving, mixing, spacing

1 Introduction

High quality pedagogy is an essential goal for MOOCs. There are few barriers to students moving between courses, and the expectations are also that online learning platforms will take advantage of their greater freedom to innovate than many education reform movements in traditional schools.

One way to complement the practical experience of quality instructors is to synthesize and apply insights from scientific research. The nature of such work is produce insights that people's direct experience is unlikely to uncover. This paper considers how research from cognitive science can improve learning in MOOCs. The following consider educational implications of cognitive science more generally. [1] is an Institute of Education Sciences practice guide that is short, available online, constructed by an expert panel, and peer-reviewed. Books include [2], which is targeted at university instructors, [3] is for a general audience and K-12 teachers, and [4] focuses on multimedia learning for both K-16 education and corporate training.

This paper follows the approach taken in the reviews above in selecting practical principles from a broad review and synthesis of literature in cognitive science. This includes publications of basic research and controlled laboratory experiments, as well as studies with educational materials and K-12 and university students from K-12 and university students – which are directly relevant to lessons in current MOOCs.

The principles are selected to target key challenges in online learning, like ensuring learners remain engaged and active even without a physical community, promoting deep understanding rather than superficial memory, and supporting students in being strategic and independent learners, even without much direct feedback.

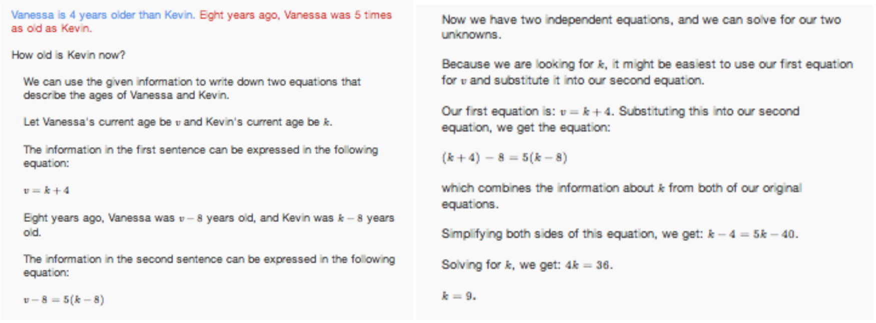
The principles specifically focus on how to appropriately prompt students to answer questions and provide explanations, *before*, *during*, and *after* watching instructional videos or engaging in exercises. It is a common intuition that students learn when they are *given* comprehensive knowledge: MOOCs deliver high-quality online videos with cogent explanations, and include practice exercises like that in Figure 1, accompanied by clear answers and solutions. However, there is substantial evidence that students can learn far more by trying to *answer* questions themselves (than by receiving the answers), or by being pushed to construct explanations (rather than provided with them), which will be discussed in the following sections.

2 Context of application: example video and exercise

Each principle for adding question prompts is targeted at the grain size of an online *module* – a short, self-contained batch of information like a video or exercise.

The principles are abstract in that they can improve learning from a range of online videos and exercises, but to provide concrete and actionable insight they are illustrated through application to specific examples of a video and exercise.

The example video is a three minute Udacity.com video from an introductory statistics course (http://tiny.cc/examplevideo): It explains what the normal distribution is, and how the area under its curve corresponds to the probability of observing certain sampled observations from a population.



Vanessa is 4 years older than Kevin. Eight years ago, Vanessa was 5 times as old as Kevin.

How old is Kevin now?

We can use the given information to write down two equations that describe the ages of Vanessa and Kevin.

Let Vanessa's current age be v and Kevin's current age be k.

The information in the first sentence can be expressed in the following equation:

$v = k + 4$

Eight years ago, Vanessa was $v - 8$ years old, and Kevin was $k - 8$ years old.

The information in the second sentence can be expressed in the following equation:

$v - 8 = 5(k - 8)$

Now we have two independent equations, and we can solve for our two unknowns.

Because we are looking for k, it might be easiest to use our first equation for v and substitute it into our second equation.

Our first equation is: $v = k + 4$. Substituting this into our second equation, we get the equation:

$(k + 4) - 8 = 5(k - 8)$

which combines the information about k from both of our original equations.

Simplifying both sides of this equation, we get: $k - 4 = 5k - 40$.

Solving for k, we get: $4k = 36$.

$k = 9$.

Fig. 1. Example math exercise from Khan Academy: http://tiny.cc/exampleexercise

The example exercise is shown in Figure 1, an algebra word problem from Khan Academy's collection of mathematics exercises at www.khanacademy.org/exercisedashboard. These share a common format. Only the problem statement is shown at first (blue & red text in Figure 1). Students can submit an answer for feedback or request a hint at any point. They only move onto the next problem when they are correct, but each hint request reveals the next step in a worked example solution – which ultimately gives the answer as its final step.

3 Adding questions before, during, and after videos & exercises

Questions or prompts to generate explanations can be added in at least three ways to online modules: *pre*-module (immediately preceding or in the very beginning of a video/exercise, preceding the presentation of content), *intra*-module (popping up in a video or emphasized as an activity by the instructor, embedded into the steps of an exercise), or *post*-module (following the student's engagement with a video/exercise).

3.1 Pre-Module: Framing Questions

Even before learners are presented with information in a video or exercise, prompting them to consider *framing questions* can make them more motivated to learn, as well as help them connect a module's content to their existing knowledge, and understand how they can apply it to future problems.

In contrast to delivering a traditional sequence of *subject-focused* videos & exercises (which touch on a succession of topics students may struggle to relate), *problem-based learning* [5] frames videos & exercises as the knowledge needed to solve particular problems and answer previously articulated questions. For example, a problem-based learning version of an introductory statistics course [6] would precede lessons with a keen emphasis on what problems the lesson would teach students how to solve, rather than a typical focus on the specific facts and concepts in each lesson.

Examples of pre-module framing questions are shown in Table 1.

Table 1. Examples of Framing Questions that could precede videos and exercises.

Udacity video on the normal distribution	Khan Academy algebra math exercise
Before a video, a page with a Framing Question can be presented: "*Explain what you already know about normal distributions*." *"What is a normal distribution useful for?"* Instructors can also introduce a fixed time delay (e.g. 10 seconds), a required text response, or a strong emphasis on a Framing Question at the start of a video.	If you are only told about the relationships between two people's ages, what kind of math is useful for figuring out actual ages? The guiding question to keep in mind for this exercise is: "How can you convert word problems into algebra expressions?"

The motivational benefit is in greater excitement to learn in order to solve a problem, rather than learn to memorize and be tested. The cognitive benefit arises in part by getting learners to activate their existing knowledge, so they connect new information to well-established ideas. Prompts to explain a fact can be largely unsuccessful, but still increase how much is learned once a lesson is presented [7]. [8]

showed that students were mostly unsuccessful when asked to solve a problem related to calculating variability, but that having tried to solve this problem changed *what* they learned from a subsequent lesson. Compared to other students who received alternative instruction without this framing question or problem, these students were better able to apply what they learned in subsequent lessons to new situations.

Developing Framing Questions. To generate framing questions for a particular resource, an instructor can ask:

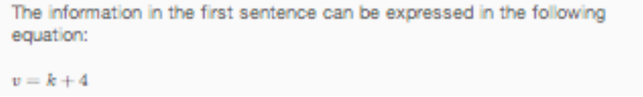
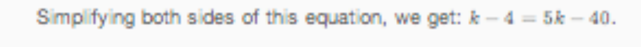
- "What questions should students be able to answer after watching this video, that they can't right now?"
- "What problems do I think they should be able to solve afterwards, that they would have struggled with before?"

3.2 Intra–Module: Reflection Questions

Typically, instruction is seen as *providing* learners with answers or *giving* them explanations. But extensive work in cognitive science, education, and intelligent tutoring has shown that giving learners the right prompts to self-generate explanations can be *more* effective than giving students explanations [9] [10]. This provides empirical insight into how and when "teaching is the best way to learn". Without changing the content of online videos and exercises, MOOCs can improve learning by appropriately embedding questions and prompts for learners to provide explanations.

Videos in MOOCs already have the functionality to pop-up short multiple choice exercises, which could be used to present questions that are more conceptual and that allow open-ended responses. Solutions to exercises can be split up into multiple lines, and have questions and prompts with text boxes to type answers embedded inline. Examples are shown in Table 2.

Table 2. Examples of how Reflection Questions could be embedded in videos and exercises.

Udacity video on normal distribution	Khan Academy algebra math exercise
Explain what the video has talked about so far. (@1:35) What are you thinking about right now? Just say it out loud. (@ 2:15)	 The information in the first sentence can be expressed in the following equation: $v = k + 4$ Do you see why this step makes sense or is justified?  Simplifying both sides of this equation, we get: $k - 4 = 5k - 40$. What step do you think is coming next?

There is substantial evidence that learners' understanding is improved by prompts to explain out loud the meaning of what they are learning or say out loud what they are thinking [9] – although studies typically ensure learners are not confused by the sudden appearance of these prompts. Asking learners to explain *why* particular facts are true or answers are correct has been shown to help them understand key principles and generalizations [11]. [12] shows that anticipating next steps in a solution and making predictions about what will be discussed next leads to a better understanding of how and where to use what they are learning about, and provides implicit feedback as the video continues or solution is revealed.

Developing Reflection Questions. In addition to examining the methods of the studies cited above, the Institute of Education Sciences practice guide [1] provides a reference of effective question stems: E.g., why, why-not, how, what-if, how does X compare to Y, what is the evidence for X?

An instructor can use a list of these stems to generate and insert question or explanation prompts throughout an instructional video or an exercise's solution.

3.3 Post–Module Memory Practice Questions

Questions that target information from a past video or exercise are common in MOOCs, but often do not realize their potential for *Memory Practice*. One reason is that they are often designed to *assess* learning without attention towards *improving* it. [13] shows that simply asking students to recall what they read in a science passage (an open ended prompt that is not common in testing, but encourages Memory Practice) greatly improved memory a week later – outperforming students who read the passage *three more times*, or made elaborate concept maps. Post-module prompts for this paper's current examples might include "Write down the main points from that video." or "Explain the method you used to solve these exercises."

In fact, MOOCs often do include post-module questions designed to help students revisit content – such as review questions or practice exercises. However, these may not successfully produce Memory Practice if they occur so soon after a module that a learner can answer using rote memory. [14] provides an extensive review of how to ensure post-module questions are beneficial, so that Memory Practice helps learners generate the meaningful cues and connections to other concepts that are needed to remember over the long-term.

For example, simply *spacing* practice exercises improves long-term retention (although benefits are deceptively absent in the *short-term*), and learning is even further improved by *interleaving* or *mixing* problems and concepts that students frequently confuse [15]. For example, a typical practice sequence might be 12 problems of type A, then 12 of type B, and 12 of type C. But it can be better for deep, lasting learning to practice [6 A, 4 B, 2 C], [4 A, 6 B, 2 C], and [2 A, 4 B, 6 C]. Often, however, students and instructors may assume that the more challenging learning in the *mixed* condition means that it is a poorer strategy and abandon it – even though it produces larger and lasting benefits *without any increase* in the number of problems [15]. Ironically, the same studies that empirically show the advantages of Memory Practice also find that students expect typical study strategies to help more [13] [14].

Conclusion

This paper considered how to improve learning in MOOCs by adding question & explanation prompts before, during, and after online videos and exercises. This is not to say that MOOCs *never* incorporate questions into instruction as advised – this is unlikely given the diversity of online instruction. Scientific principles for learning can be used to design novel instruction *or* to support *benchmarking* – to identify which of the vast set of instructional strategies are supported by cognitive science. Moreover, consulting and working with cognitive scientists (to embed practical experiments and design measures of learning) allows MOOCs to maximize learning by tailoring general learning principles to specific courses and lessons. Collaborations like these between instructions and scientists can provide the best outcomes for students.

References

1. Pashler, H., Bain, P., Bottge, B., Graesser, A., Koedinger, K., McDaniel, M., Metcalfe, J.: Organizing Instruction and Study to Improve Student Learning (NCER 2007-2004). Washington, DC: Institute of Education Sciences, U.S. Department of Education (2007)
2. Ambrose, S. A., Bridges, M. W., DiPietro, M., Lovett, M. C., Norman, M. K.: How learning works: Seven research-based principles for smart teaching. Jossey-Bass (2010)
3. Willingham, D. T.: Why Don't Students Like School. Jossey-Bass (2010)
4. Clark, R. C., & Mayer, R. E.: E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning. Pfeiffer (2004)
5. Hmelo-Silver, C. E.: Problem-based learning: What and how do students learn? Educational Psychology Review, 16(3), 235-266 (2004)
6. Boyle, C. R.: A problem-based learning approach to teaching biostatistics. Journal of Statistics Education, 7(1) (1999)
7. Needham, D. R., Begg, I. M.: Problem-oriented training promotes spontaneous analogical transfer: Memory-oriented training promotes memory for training. Memory & Cognition 19, 543–557 (1991)
8. Schwartz, D.L., Martin, T.: Inventing to Prepare for Future Learning: The Hidden Efficiency of Encouraging Original Student Production in Statistics Instruction. Cognition and Instruction, 22(2), 129-184 (2004)
9. Fonseca, B. Chi, M.T.H.: The self-explanation effect: A constructive learning activity. In: Mayer, R. & Alexander, P. (Eds.), The Handbook of Research on Learning and Instruction (pp. 270-321). New York, USA: Routledge Press (2011)
10. Mazur, E.: Farewell, Lecture? Science (2009)
11. Williams, J. J., Walker, C. M., Lombrozo, T.: Explaining increases belief revision in the face of (many) anomalies. In: N. Miyake, D. Peebles, & R. P. Cooper (Eds.), Proceedings of the 34th Annual Conference of the Cognitive Science Society (pp. 1149-1154). Austin, TX: Cognitive Science Society (2012)
12. Renkl, A.: Learning from Worked-Out Examples: A Study on Individual Differences. Cognitive Science, 21(1), 1–29 (1997)
13. Karpicke, J. D., Blunt, J. R.: Retrieval practice produces more learning than elaborative studying with concept mapping. Science, 331(6018), 772-775 (2011)
14. Roediger, H. L., Putnam, A. L., Smith, M. A.: Ten benefits of testing and their applications to educational practice. In: J. Mestre & B. Ross (Eds.), Psychology of learning and motivation: Cognition in education (pp. 1-36). Oxford: Elsevier (2011)
15. Rohrer, D., Pashler, H.: Increasing retention without increasing study time. Current Directions in Psychological Science, 16, 183-186 (2007)

Measurably Increasing Motivation in MOOCs

Joseph Jay Williams[1], Dave Paunesku[2], Benjamin Haley[3], Jascha Sohl-Dickstein[2,4]

[1] U.C. Berkeley, [2]Stanford University, [3]Northwestern University, [4]Khan Academy
United States
joseph_williams@berkeley.edu, paunesku@stanford.edu, benjamin.haley@gmail.com,
jascha@khanacademy.org

Abstract. A key challenge in online learning is keeping students motivated. We report an experiment that added motivational messages to students solving mathematics problems on the KhanAcademy.org platform. By simply adding sentences above the text of a math problem, students attempted (successfully) a greater number of problems, were more likely to acquire exercise proficiencies, and even solved a larger proportion of attempted problems correctly. The key feature for producing these measurably improved outcomes was in using messages that emphasized that intelligence is malleable – e.g., “Remember, the more you practice the smarter you become!”. Control conditions that provided neutral science facts or even positive messages – e.g., “This might be a tough problem, but we know you can do it.” – were not as effective. There are many pedagogical strategies that instructors of online courses might hypothesize will increase motivation; these findings underscore the value in empirically testing such predictions, using the unique data that is now available in MOOCs.

Keywords: motivation, MOOCs, learning, mindset

Controlled experiments on millions of students to personalize learning

Eliana Feasley[1,2], Chris Klaiber[1], James Irwin[1], Jace Kohlmeier[1], Jascha Sohl-Dickstein[1,3]
[1]Khan Academy, [2]UT Austin, [3]Stanford
United States
{eliana, chris, james, jace, jascha}@khanacademy.org

Abstract. Khan Academy is a personalized learning resource that enables students to watch educational videos and answer questions across a variety of levels of mathematics and other subjects. With over one billion problems solved, Khan Academy has a massive dataset from which to draw evidence and make inferences about student learning behaviors. Our goal is to use this unprecedented quantity of data to learn what content each student will benefit the most from seeing, and to present it to them. Towards this goal, we have run more than one hundred massive controlled experiments, evaluating hypotheses about learning.

We focus here on personalizing the learning experience by using student responses to assessment items to adaptively suggest new content. We discuss the metrics by which we measure student improvement and the tradeoffs that occur when increased exercise difficulty reduces student engagement. We further discuss personalizing content such as exercise or video suggestions, and measuring student responses to such interventions. Leveraging massive data to personalize learning is one of the greatest promises of online education, and this work represents first steps towards fulfilling that promise for millions of users worldwide.

Keywords: personalized learning, data mining, machine learning, massive data, Khan Academy

Analysis of video use in edX courses

Daniel T. Seaton, Albert J. Rodenius, Cody A. Coleman, David E. Pritchard, Isaac Chuang
Massachusetts Institute of Technology
United States
{dseaton, albertr, colemanc, dpritch, ichuang}@mit.edu

Abstract. In Massive Open Online Courses (MOOCs), online videos serve as the equivalent of lectures found in their traditional on-campus courses. Across a number of courses offered by edX in the Fall of 2012, the number of unique videos watched shows bimodal student engagement similar to ``attendance'' of large-lecture on-campus courses; only half the participants are watching the majority of course videos. The overall scale of MOOC populations still allows for meaningful measurements of video activity, while also providing a tremendous opportunity to experiment with methods of improving engagement of those participants showing low video use. We present preliminary analyses of the nature of video engagement through both the fraction of videos viewed over the course and the detection of convergent activity (``hot spots'') in the collective pause and play interactions within each video. We discuss our results in the context of improving video content, as well as a new video annotation tool being integrated into assessment items.

Keywords: MOOC, Video, Online, Analytics

Exploring Possible Reasons behind Low Student Retention Rates of Massive Online Open Courses: A Comparative Case Study from a Social Cognitive Perspective

Yuan Wang
Columbia University
United States
elle.wang@columbia.edu

Abstract. Massive Open Online Courses (MOOCs) have been widely lauded by the press since its fairly recent inception. Besides its wide popularity among learners worldwide, the majority of MOOCs still present challenges with steep dropout rates in spite of their promising enrollment numbers. While enjoying various benefits MOOCs brings along, learners apparently face new challenges. This paper intends to explore possible reasons behind this phenomenon from a social cognitive perspective by analyzing and comparing the same subject content taught in both the traditional face-to-face setting and on a MOOC-based platform.

Based on past research and theories including both the larger distance learning fields as well as recent MOOC-specific ones, three areas, namely, the lack of self-efficacy, self-regulation, and self-motivators are identified to help present an exploratory framework in interpreting findings of this study. Although far from all encompassing, this exploratory framework attempts to enhance our understanding of distinct challenges MOOC learners as well as MOOC designers face.

Keywords: MOOCs, Distance Learning, Student Retention Rate, Sustainability of Learning.

Using EEG to Improve Massive Open Online Courses Feedback Interaction

Haohan Wang, Yiwei Li, Xiaobo Hu, Yucong Yang, Zhu Meng, Kai-min Chang

Language Technologies Institute
School of Computer Science
Carnegie Mellon University

Abstract. Unlike classroom education, immediate feedback from the student is less accessible in Massive Open Online Courses (MOOC). A new type of sensor for detecting students' mental states is a single-channel EEG headset simple enough to use in MOOC. Using its signal from adults watching MOOC video clips in a pilot study, we trained and tested classifiers to detect when the student is confused while watching the course material. We found weak but above-chance performance for using EEG to distinguish when a student is confused or not. The classifier performed comparably to the human observers who monitored student body language and rated the students' confusion levels. This pilot study shows promise for MOOC-deployable EEG devices being able to capture tutor relevant information.

Keywords: MOOC, EEG, confuse, feedback, machine learning

1 Introduction

In recent years, there is an increasing trend towards the use of Massive Open Online Courses (MOOC), and it is likely to continue [1]. MOOC can serve millions of students at the same time, but it has its own shortcomings. In [2], Thompson studied post-secondary students who had negative attitudes toward correspondence-based distance education programs. The results indicate that lack of immediate feedback and interaction are two problems with long-distance education. Current MOOC can offer interactive forums and feedback quizzes to help improve the communication between students and professors, but the impact of the absence of a classroom is still being hotly debated. As also discussed in [3], indicates the lack of feedback is one of the main problems for student-teacher long distance communication.

There are many gaps between online education and in-class education [4] and we will focus on one of them: detecting students' confusion level. Unlike in-class education, where a teacher can judge if the students understand the materials by verbal inquiries or noticing their body language (e.g., furrowed brow, head scratching, etc.), immediate feedback from the student is less accessible in long distance education. We address this limitation by using electroencephalography (EEG) input from a commercially available device as evidence of students' mental states.

The EEG signal is a voltage signal that can be measured on the surface of the scalp, arising from large areas of coordinated neural activity manifested as synchronization (groups of neurons firing at the same rate) [5]. This neural activity varies as a function of development, mental state, and cognitive activity, and the EEG signal can measurably detect such variation. Rhythmic fluctuations in the EEG signal occur within several particular frequency bands, and the relative level of activity within each frequency band has been associated with brain states such as focused attentional processing, engagement, and frustration [6-8], which in turn are important for and predictive of learning [9].

The recent availability of simple, low-cost, portable EEG monitoring devices now makes it feasible to take this technology from the lab into schools. The NeuroSky "MindSet," for example, is an audio headset equipped with a single-channel EEG sensor [10]. It measures the voltage between an electrode that rests on the forehead and electrodes in contact with the ear. Unlike the multi-channel electrode nets worn in labs, the sensor requires no gel or saline for recording and therefore requires much less expertise to position. Even with the limitations of recording from only a single sensor and working with untrained users, a previous study [11] found that the MindSet distinguished two fairly similar mental states (neutral and attentive) with 86% accuracy. MindSet has been used to detect reading difficulty [12] and human emotional responses [13] in the domain of intelligent tutoring systems.

A single-channel EEG device headset currently costs around $99-149 USD, which would be a cost deterant to the free service of MOOC. We suggest that MOOC providers (e.g., Coursera, edX) supply EEG devices to a select group of students. In return, MOOC providers would get feedback on students' EEG brain activity or confusion levels while students watch the course materials. These objective EEG brain activities can be aggregated and augment subjective rating of course materials to provide a simulation of real world classroom responses, such as when a teacher is given feedback from an entire class. Then teachers can improve video clips based on these impressions. Moreover, even though an EEG headset is a luxury device at the moment, the increasing popularity of consumer-friendly EEG devices may one day make it a house-hold accessory like audio headsets, keyboards and mice. Thus, we are hopeful of seeing our proposed solution come to fruition as the market for MOOC grows and the importance of course quality and student feedback increases.

To assess the feasibility of collecting useful information about cognitive processing and mental states using a portable EEG monitoring device, we conducted a pilot study with college students watching MOOC video clips. We wanted to know if EEG data can help distinguish among mental states relevant to confusion. If we can do so by better than chance, then these data may contain relevant information that can be decoded more accurately in the future. Thus, we address two questions:

1. Can EEG detect confusion?
2. Can EEG detect confusion better than human observers?

The rest of this paper is organized as follows. Section 2 describes the experiment design. Section 3 and 4 answers the two research questions, respectively. Finally, Section 5 concludes and suggests future work.

2 Experiment Design

In a pilot study, we collected EEG signal data from college students while they watched MOOC video clips. We extracted online education videos that are assumed not to be confusing for college students, such as videos of introduction of basic algebra or geometry. We also prepare videos that are assumed to confuse a normal college student if a student is not familiar with the video topics like Quantum Mechanics, and Stem Cell Research[1]. We prepared 20 videos, 10 in each category. Each video was about 2 minutes long. We chopped the two-minute clip in the middle of a topic to make the videos more confusing.

We collected data from 10 students. One student was removed because of missing data due to technical difficulties. An experiment with a student consisted of 10 sessions. We randomly picked five videos of each category and randomized the presentation sequence so that the student could not guess the predefined confusion level. In each session, the student was first instructed to relax their mind for 30 seconds. Then, a video clip was shown to the student where he/she was instructed to try to learn as much as possible from the video. After each session, the student rated his/her confusion level on a scale of 1-7, where 1 corresponded to the least confusing and 7 corresponded to the most confusing. Additionally, there were three student observers watching the body-language of the student. Each observer rated the confusion level of the student in each session on a scale of 1-7. The conventional scale of 1-7 was used. Four observers were asked to observe 1-8 students each, so that there was not an effect of observers just studying one student.

The students wore a wireless single-channel MindSet that measured activity over the frontal lobe. The MindSet measures the voltage between an electrode resting on the forehead and two electrodes (one ground and one reference) each in contact with an ear. More precisely, the position on the forehead is Fp_1 (somewhere between left eye brow and the hairline), as defined by the International 10-20 system [14]. We used NeuroSky's API to collect the EEG data.

3 Can EEG detect confusion?

We trained Gaussian Naïve Bayes classifiers to estimate, based on EEG data, the probability that a given session was confusing rather than not confusing. We chose this method (rather than, say, logistic regression) because it is generally best for problems with sparse (and noisy) training data [15].

To characterize the overall values of the EEG signals while the students watch the 2 minute video, we computed their means over the interval. To characterize the temporal profile of the EEG signal, we computed several features, some of them typically used to measure the shape of statistical distributions rather than of time series: minimum, maximum, variance, skewness, and kurtosis. However, due to the small number of data points (100 data points for 10 subjects, each watching 10 videos), inclusion of

[1] http://open.163.com/

those features tends to overfit the training data and results in poor classifier performance. As a result, we used the means as the classifier features for the main analysis. **Table 1** shows the classifier features.

Table 1. Classifier features

Features	Description	Sampling rate	Statistic
Attention	Proprietary measure of mental focus	1 Hz	Mean
Meditation	Proprietary measure of calmness	1 Hz	Mean
Raw	Raw EEG signal	512 Hz	Mean
Delta	1-3 Hz of power spectrum	8 Hz	Mean
Theta	4-7 Hz of power spectrum	8 Hz	Mean
Alpha1	Lower 8-11 Hz of power spectrum	8 Hz	Mean
Alpha 2	Higher 8-11 Hz of power spectrum	8 Hz	Mean
Beta1	Lower 12-29 Hz of power spectrum	8 Hz	Mean
Beta 2	Higher 12-29 Hz of power spectrum	8 Hz	Mean
Gamma1	Lower 30-100 Hz of power spectrum	8 Hz	Mean
Gamma2	Higher 30-100 Hz of power spectrum	8 Hz	Mean

To avoid overfitting, we used cross validation to evaluate classifier performance. We trained student-*specific* classifiers on a single student's data from all but one stimulus block (e.g., one video), tested on the held-out block (e.g., all other videos), performed this procedure for each block, and averaged the results to cross-validate accuracy within reader. We trained *student-independent* classifiers on the data from all but one student, tested on the held-out student, performed this procedure for each student, and averaged the resulting accuracies to cross-validate across students.

We use two ways to label the mental states we wish to predict. One way is the *pre-defined* confusion level according to the experiment design. Another way is the *user-defined* confusion level according to each user's subjective rating.

Detect pre-defined confusion level. We trained and tested classifiers for pre-defined confusion. Student-specific classifiers achieve a classification accuracy of 67% and a kappa statistic of 0.34, whereas student-independent classifiers achieve a classification accuracy of 57% and a kappa statistic of 0.15. Both classifier performances were statistically significant better than a chance level of 0.5 ($p < 0.05$). **Fig. 1a)** plots the classifier accuracy for each student. **Fig. 1a)** shows that both student-specific classifiers and student-independent classifiers performed significantly above chance in 6 out of 9 students.

Detect user-defined confusion level. We also trained and tested classifiers for student-defined confusion. Since students have different sense of confusing, we mapped the seven scale self-rated confusion level into a binary label, with roughly equal number of cases in the two classes. A middle split is accomplished by mapping scores less than or equal to the median to "not confusing" and the scores greater than the median are mapped to "confusing". Furthermore, we used random undersampling of the larger class(es) to balance the classes in the training data. We performed the sampling 10 times to limit the influence of particularly good or bad runs and obtain a stable measure of classifier performance.

Student-specific classifiers achieve a classification accuracy of 57% and a kappa statistic of 0.13, whereas student-independent classifiers achieve a classification accuracy of 51% and a kappa statistic of -0.04. The student-specific classifier performance was statistically significant and better than a chance level of 0.5 ($p < 0.05$), but not the student-independent classifier. **Fig. 1b)** plots the accuracy for each student. **Fig. 1b)** shows that the student-specific classifier performed significantly above chance for 5 out of 9 students and student-independent classifier performed significantly above chance for 2 out of 9 students.

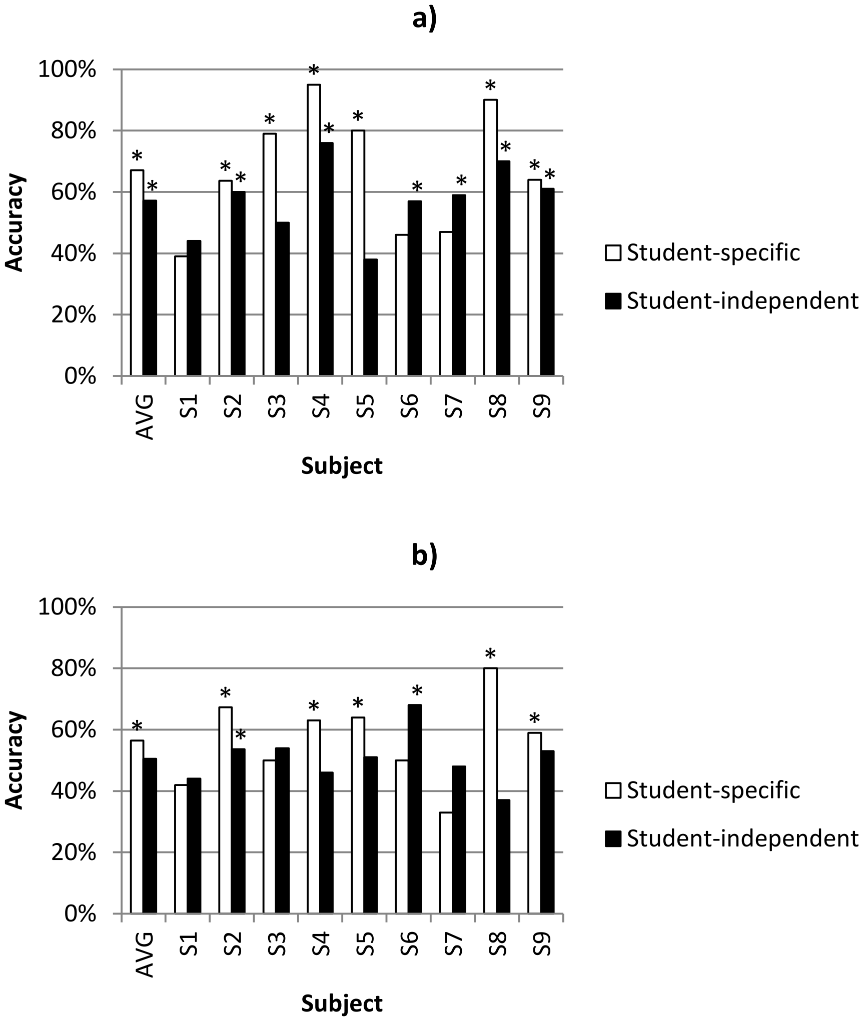


Fig. 1. Detect a) predefined, and b) user-defined confusion level

4 Can EEG detect confusion better than human observers?

To determine if EEG can detect confusion better than human observers of body language, we compared the scores from the observers, the classifier, and the students, with the label of videos. For each student, we used the average scores of the observers as the ‘observer rating’. We used the classifier trained in Section 3 to predict predefined confusion level and linearly mapped the classifier’s estimate of class probability (0-100%) to a scale of 1-7 and labeled it as the ‘classifier rating’.

Fig. 2 shows the scatter plot of a) student vs. observer rating, and b) student vs. classifier rating. The classifier rating had a low, but positive correlation (0.17) with the students’ rating, while the observer rating had a low, but positive correlation of (0.17) with the students’ rating. This shows that the classifier performed comparably to the human observers who monitored student body language and rated the students’ confusion levels.

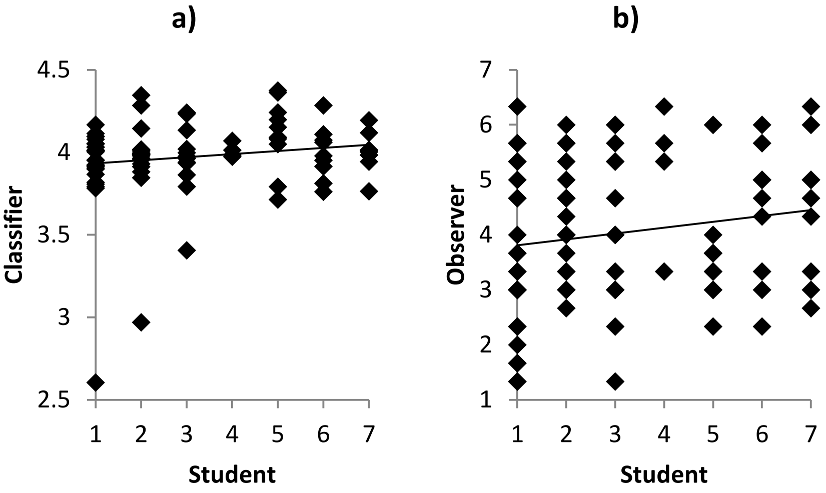


Fig. 2. Scatter plot of a) classifier vs. student rating, and b) observer vs. student rating

5 Conclusions and Future Work

In this paper, we described a pilot study, where we collected students’ EEG brain activity while they watched MOOC video clips. We trained and tested classifiers to detect when a student was confused. We found weak but above-chance performance for using EEG to distinguish whether a student is confused. The classifier performed comparably to the human observers who monitored student body language and rated the students’ confusion levels.

Since the experiment was based on a class project run by a group of graduate students, there were many limitations to the experiment. We now discuss the major limitations and how we plan to address them in future work.

One of the most critical limitations is the definition of experimental construct. Specifically, our pre-defined "confusing" videos could be confounded. For example, a student may not find a video clip on Stem Cell to be confusing when the instructor clearly explains the topic. Also, the predefined confusion level may be confounded with increased mental effort / concentration. To explore this issue, we examined the relationship between the predefined confusion level and the subjective user-defined confusion level. The students' subjective evaluation of the confusion level and our predefined label has a modest correlation of 0.30. Next, we performed a feature selection experiment among all combinations of 11 features; we used cross validation through all the experiments and sorted the combinations according to accuracy. Then we found that the user-specific model Theta signal played an important role in all the leading combinations. Theta signal corresponds to errors, correct responses and feedback, suggesting the experimental construct is indeed related to confusion.

Another limitation is due to the lack of psychological professionalism. For example, the observers in our experiment were not formally trained. As a result, the current scheme allowed each observer to interpret a student's confusion level based on his/her own interpretation. A precise labeling scheme would yield more details that could be compared among raters and, thereby, improve our rating procedure.

Another limitation is the scale of our experiment as we only performed the experiments with 10 students, and each student only watched 10 two-minute video clips. The limited amount of data points prevents us from drawing any strong conclusions about the study. We hope to scale up the experiment and collect more data.

Finally, this pilot study shows positive, but weak classifier performance in detecting confusion. The weak classifier performance may have many false-alarms and thereby frustrate a student. In addition, a student may not be willing to share their brain activity data due to privacy concerns. We are hopeful that the classifier accuracy can be improved once we conduct a more rigorous experiment, by increasing the study size, and improve the classifier with better feature selection and by applying denoising techniques to improve signal-to-noise ratio. Lastly, the classifiers are supposed to help students, so the students should be able to choose not to use EEG if they think the device is hindering.

Acknowledgments. This work was supported by the National Science Foundation under Cyberlearning Grant IIS1124240. The opinions expressed are those of the authors and do not necessarily represent the views of the Institute, or the National Science Foundation. We thank Jessica Nelson for help with experiment design, Donna Gates for preparation of the manuscript, and the students, educators, and LISTENers who helped generate, collect, and analyze our data.

Reference

1. Allen, I.E., Seaman, J., *Going the Distance: Online Education in the United States, 2011*, 2011.
2. Thompson, G., *How Can Correspondence-Based Distance Education be Improved?: A Survey of Attitudes of Students Who Are Not Well Disposed toward Correspondence Study.* The Journal of Distance Education, 1990. **5**(1): p. 53-65.
3. Shute, V., et al. *Assessment and learning in intelligent educational systems: A peek into the future*. in *Proceedings of the 14th International Conference on Artificial Intelligence in Education Workshop on Intelligent Educational Games*. 2009. Brighton, UK.
4. Vardi, M.Y., *Will MOOCs destroy academia?*, in *Communications of the ACM*2012. p. 5.
5. Niedermeyer, E., Fernando H. Lopes da Silva, F. H., *Electroencephalography: basic principles, clinical applications, and related fields*2005: Lippincott Williams & Wilkins.
6. Marosi, E., et al., *Narrow-band spectral measurements of EEG during emotional tasks.* International Journal of Neuroscience, 2002. **112**(7): p. 871-891.
7. Lutsyuk, N.V., E.V. Éismont, and V.B. Pavlenko, *Correlation of the characteristics of EEG potentials with the indices of attention in 12- to 13-year-old children.* Neurophysiology, 2006. **38**(3): p. 209-216.
8. Berka, C., et al., *EEG correlates of task engagement and mental workload in vigilance, learning , and memory tasks.* Aviation, Space, and Environmental Medicine, 2007. **78 (Supp 1)**: p. B231-244.
9. Baker, R., et al., *Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive-affective states during interactions with three different computer-based learning environments.* International Journal of Human-Computer Studies, 2010. **68**(4): p. 223-241.
10. NeuroSky, *Brain wave signal (EEG)*, 2009, Neurosky, Inc.
11. NeuroSky, *NeuroSky's eSense™ meters and dtection of mntal sate*, 2009, Neurosky, Inc.
12. Mostow, J., K.M. Chang, and J. Nelson. *Toward exploiting EEG input in a Reading Tutor*. in *15th International Conference on Artificial Intelligence in Education*. 2011. Auckland, New Zealand: Lecture Notes in Computer Science.
13. Crowley, K., et al., *Evaluating a brain-computer interface to categorise human emotional response* in *10th IEEE International Conference on Advanced Learning Technologies*2010: Sousse, Tunisia. p. 276-278.
14. Jasper, H.H., *The ten-twenty electrode system of the International Federation.* Electroencephalography and Clinical Neurophysiology, 1958. **10**: p. 371-375.
15. Ng, A.Y. and M.I. Jordan. *On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes* in *Advances in Neural Information Processing Systems* 2002. MIT Press.

Collaborative Learning in Geographically Distributed and In-person Groups

René F. Kizilcec

Department of Communication, Stanford University, Stanford CA 94305
kizilcec@stanford.edu

Abstract. Open online courses attract a diverse global audience of learners, many of whom might not be self-directed autodidacts with the necessary web competencies to reap the full benefits of such courses. Most of these learners would benefit from increased guidance on how to use MOOCs to enhance their learning. One potential area for guidance is in group collaboration where learners form teams to collaboratively work on assignments. Despite the global scope of these courses, a large proportion of learners live within relatively close proximity of each other, such that in-person collaboration is a feasible option. However, geographically distributed groups of learners are more likely to bring diverse viewpoints to the discussion than learners who live close to each other. Research suggests that the diversity of viewpoints in a group positively affects the quality of collaboration and outcomes. This paper reviews the literature on the feasibility of assigning local groups for collaboration and proposes concrete research directions.

1 Introduction

An increasing number of educators use online, asynchronous computer-mediated communication tools to create massive open online courses (MOOCs). These virtual classrooms attract a global audience of learners (Fig. 1) who join these courses for various reasons, including earning a certificate for completing the course or personal enrichment. The global and massive scale of these courses make them a melting pot for diverse ideas and perspectives: the learner population varies considerably in demographics, cultural background, language skills, personality, motivation, and prior knowledge.

Potentially the most important scholarly question in the midst of the rapid proliferation of open online courses is how learning can be enhanced with MOOCs. No simple answer can suffice, but it is clear that understanding the learner population is critical for developing strategies to foster learning. Borrowing a term from Lévi-Strauss [1], the online learner can be understood as a *bricoleur*–a handy-man or jack-of-all-trades–who cobbles together ways to learn from the plethora of online learning resources. The danger with this notion of the learner is that it is probably over-optimistic, given that many learners are not autodidacts or not "MOOC-ready" in other ways, e.g. not technologically adept. Hence, to ensure equal opportunities to learn, we need to provide guidance to learners to become skilled *bricoleurs* and continuously support them in their *bricolage* learning endeavor.

2 Collaborative Learning

Small group collaboration in and around MOOCs is a particularly fertile ground for increased guidance. The literature on computer-supported collaborative learning can provide theoretically and empirically grounded advice on how to support group collaboration. In addition, the rapid development of the online learning space is providing opportunities for empirical research, unprecedented in scale, to test existing recommendations and investigate novel approaches to guiding group collaboration in a variety of contexts.

Many contemporary MOOCs involve group projects as part of the course, providing learners with the opportunity to collaborate with a diverse set of people and to engage in a process of knowledge building. Group characteristics affect a group's performance, satisfaction, and processes of collaborative learning.

Group formation can follow one of two philosophies: laissez-faire (self-formed) or interventionist (assigned randomly or based on certain criteria). Both approaches raise questions of how groups are selected and the kind of guidance that should be provided from the MOOC interface or other sources.

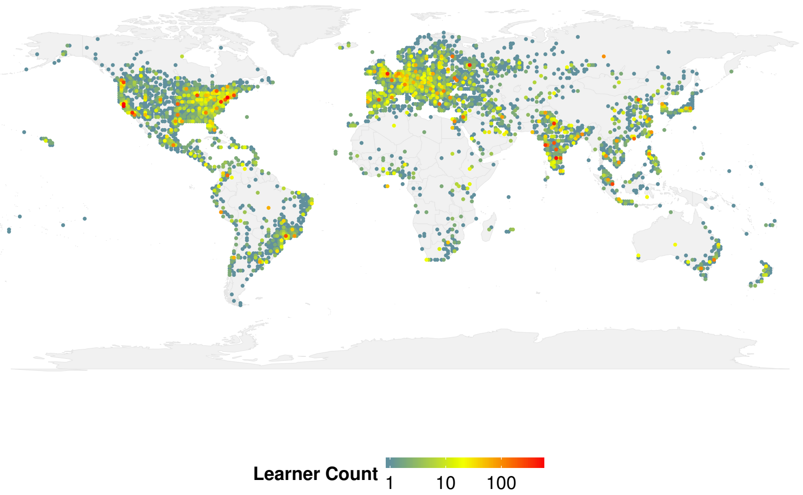
How should one form groups and guide them to encourage effective and fruitful collaboration? The remainder of the paper addresses this question. Section 3 motivates the distinction between geographically distributed and in-person groups, and presents evidence for the feasibility of assigning local groups. Section 4 reviews relevant literature on small group collaboration that can inform group assignment and guidance strategies. Section 5 proposes concrete research directions to empirically investigate strategies for group assignment and guidance, and proposes a collaboration model that combines geographical diversity and in-person collaboration. Section 6 presents concluding remarks.

3 Geographically Distributed or In-person?

Geographically distributed groups in MOOCs rely on computer-mediated communication (CMC) to work collaboratively on their project. These learners use video conferencing, and synchronous as well as asynchronous textual interfaces, such as email, instant messaging, and word processing applications with real-time collaboration. In contrast, geographic proximity can permit face-to-face (FtF) interaction. Of the two models, FtF collaboration has been associated with a significantly better learning experience in terms of the quality of group discussion and interactions compared to collaboration via asynchronous CMC [2]. This is not surprising given that FtF communication is a considerably more expressive medium than CMC.[1] However, no significant differences in learning measured by pre-post tests and self-report were found [2, 3].

[1] Interactions in immersive virtual reality are potentially more expressive than face-to-face, but the technology is not yet publicly accessible.

Fig. 1. Geographical location of active (interacted with learning materials) learners averaged over 21 MOOCs with colors representing geographical density of learners in the region. In green, yellow, and red regions, the learner population is sufficiently dense to support in-person collaboration.



In-person groups tend to be self-formed groups of friends, as geographic proximity is positively related to friendship. Such self-formed groups are subject to people's natural tendency to engage with people who are similar to themselves (homophily) [4]. The combination of homophily and the correlation between geography and demographic and other characteristics tends to make these groups even more homogeneous relative to, for instance, randomly-assigned groups. This can be a problem because collaborative learning in heterogeneous groups can be more effective than in homogeneous ones, as the wealth of alternative perspectives sparks innovative ideas [5, 6]. The research on the relationship between group members' friendship and outcomes remains split on whether collaborating with friends is beneficial [7].

The kind of guidance provided to learners partially depends on whether collaboration is in-person or computer-mediated. However, there has been no conclusive evidence that assigning groups to facilitate in-person collaboration in MOOCs is possible at a large scale. While a single MOOC attracts hundreds of thousands of learners, the feasibility of in-person collaboration relies on how many learners live close enough to fellow learners. To investigate the feasibility of in-person collaboration, geographical location data from 21 MOOCs on various topics was aggregated to produce two figures. Conclusions drawn from these data are very likely to be generalizable across MOOCs offered around the same time (late 2011 to early 2013) on MOOC platforms built around weekly video lectures and assignments.

Figure 1 illustrates the density of the active learner population on a world map.[2] Green, yellow, and red regions indicate geographical locations with sufficiently many learners to support in-person collaboration.[3]

Figure 2 illustrates the geographical density of active learners by the number of learners in the same region. At least three (five) learners live in 52% (37%) of the regions (dotted line). Moreover, due to the high learner density in a few big cities, 92% (85%) of learners live in regions with at least four (nine) other learners taking the same course (solid line). These data suggest that the distribution of learners in most parts of the world would support group assignments that facilitate in-person collaboration.

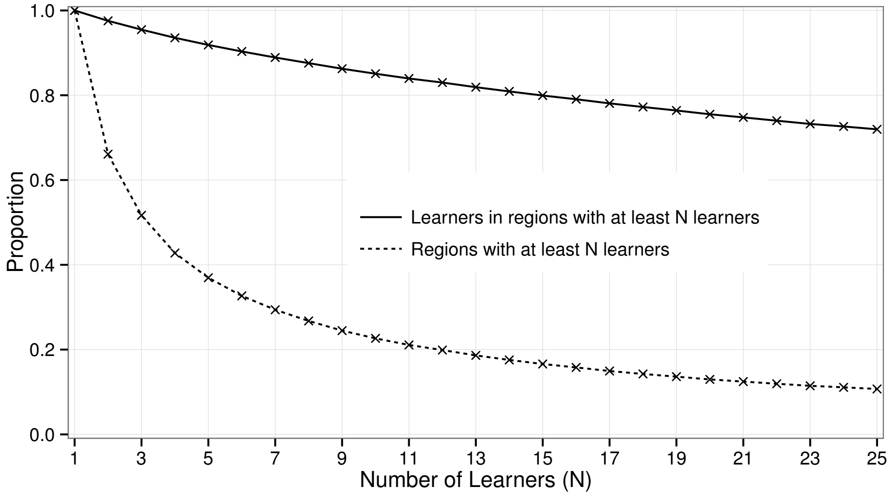
4 Relevant Literature

Scott Page's [8] work on group collaboration indicates that the diversity of viewpoints within a group is more important than the excellence of its individual members. It is reasonable to assume that people's diversity of viewpoints increases with the geographical distance between them, which would suggest that

[2] Active learners, a small subset of the enrolled learners, are defined to have used the learning materials at least once.

[3] Geographical location was determined based on users' IP address. A region is defined by all equivalent latitude/longitude coordinates rounded to zero decimal places. This definition of a region is not ideal, because the area within regions varies depending on geographical location, but it provides a rough estimate.

Fig. 2. Geographical topology of active learners (interacted with learning materials) averaged over 21 MOOCs. For 1 to 25 learners (N), the solid line illustrates the proportion of learners in regions with at least N learners and the dotted line illustrates the proportion of regions with at least N learners.



groups should be assigned with greater geographical diversity. However, there is potentially enough cultural diversity present in most major cities to assign groups with diverse viewpoints, while maintaining the geographical proximity to facilitate in-person collaboration.

Related to Page's research, Woolley and colleagues [9] report evidence for a collective intelligence in groups that has little association with the average or maximum individual intelligence of group members, but is highly correlated with the proportion of females in the group and the distribution of conversational turn-taking. While the gender distribution can be addressed by specific assignment of groups, the conversational dynamics within the group can only be influenced indirectly, for instance, by guiding group interactions technologically or with written guidelines on turn-taking. Online video conferencing tools could include timers for each participant, similar to chess clocks, to encourage balanced participation and turn-taking.

Barron's [10] findings provide further evidence that emphasizes the importance of nuanced process indicators in collaborative learning. She found indicators such as listening to proposals in group collaboration to be predictive of collaboration success, while less process-oriented measures such as group members prior achievements and how well they generated correct ideas were not correlated with positive problem-solving outcomes. Research on collaborative learning suggests that it is most effective when group members engage in rich interactions, like discussing conceptual explanations rather than providing specific answers. Thus, rich interactions can be encouraged by guiding the collaborative process

[11], for example, by providing note-taking templates that encourage certain behaviors, such as discussing conceptual explanations.

The collaboration process and how it should be guided depends on the communication medium used for collaboration. The expressiveness of the communication medium is a likely moderator of the richness of interactions [12], with FtF enabling more expressive interactions than CMC. However, advances in the learning sciences on collaborative learning with video [13] suggest that augmented CMC (augmented with tools to foster mutual awareness) can yield higher collaboration quality and learning gains than unaugmented CMC. Guidance to learners on the use of such tools, such as when and how to use them effectively, is necessary to maximize their potential benefit to learners. For instance, groups with geographically diverse members should receive guidance on several online collaboration tools, including the types of tasks that each is most suitable for and examples of how to use them effectively.

5 Research Directions

MOOCs provide researchers with a powerful platform for conducting experiments to address questions around collaborative learning in this novel context. The massive scale of these courses combined with randomized controlled field experiments can provide insights into the features of the learning environment and the kinds of guidance that can significantly enhance learning.

The effectiveness of geographically distributed compared to in-person collaboration with different models of guidance could be investigated by assigning half the project groups to maximize group members' geographic distance from each other and the other half to groups close enough to facilitate in-person collaboration. Groups could be randomly assigned to receive different guidance on collaboration strategies and technologies. Outcome measures should capture group performance (project grades and perceived learning), collaboration quality (e.g. Meier et al.'s [14] rating scheme), members' experience, and whether in-person collaboration took place for locally assigned groups. Moreover, a measure of perceived social and cultural group diversity could provide insights into the association between geographic distance and subjective group diversity, potentially an important mediator of the above outcome measures.

Beyond the question of how groups are actually assigned, the psychological implications of what learners are told about how their group members were chosen might influence their perception of the group and collaboration experience (framing effect). For example, telling learners that their collaborators were carefully chosen based on their personality and previous experience to promote productive collaboration and original ideas sets positive expectations compared to telling them that groups are randomly chosen.

An implementation that reaps the benefits of geographically distributed and in-person collaboration could be to facilitate collaboration in two steps: locally assigned groups could first collaborate in-person before connecting with a few other groups from around the world to form a larger, more distributed group that discusses the preliminary ideas and continues the collaboration online. This model of collaboration could be tested and adjusted through iterative improvement to optimize the collaboration experience.

6 Conclusion

Providing online learners with guidance, especially those who are not self-directed autodidacts, is necessary to ensure equal opportunity to learn. Group collaboration, where peers collectively solve a task or discuss an issue, is a potentially fruitful setting for increased guidance. Learning from and with peers to complement learning from the instructor is becoming increasingly important in online learning due to rapidly growing student-to-teacher ratios. It is therefore critical that collaborative learning is enhanced by providing learners with appropriate guidance.

What kind of guidance to provide will partly depend on the type of learner interaction. This paper argues that there is an important distinction between groups that have the potential for face-to-face communication and those who do not, especially as education moves out of brick-and-mortar institutions where students are all geographically accessible.

7 Acknowledgments

The author would like to thank Clifford Nass and Roy Pea for their feedback on an earlier draft of this paper and Stanford's Office of the Vice Provost for Online Learning for supporting this research.

References

1. Lévi-Strauss, C.: The savage mind. New York: Free Press. (1966)
2. Ocker, R., Yaverbaum, G.: Asynchronous computer-mediated communication versus face-to-face collaboration: Results on student learning, quality and satisfaction. Group Decision and Negotiation **8** (1999) 427–440
3. Francescato, D., Porcelli, R., Mebane, M., Cuddetta, M., Klobas, J., Renzi, P.: Evaluation of the efficacy of collaborative learning in face-to-face and computer-supported university contexts. Computers in Human Behavior **22**(2) (March 2006) 163–176
4. McPherson, M., Smith-Lovin, L., Cook, J.: Birds of a feather: Homophily in social networks. Annual Review of Sociology **27**(2001) (2001) 415–444
5. Webb, N.: Task-related verbal interaction and mathematics learning in small groups. Journal for Research in Mathematics Education **22**(5) (1991) 366–389
6. Nemeth, C.J.: Differential contributions of majority and minority influence. Psychological Review **93**(1) (1986) 23–32
7. Maldonado, H., Klemmer, S., Pea, R.: When is collaborating with friends a good idea? Insights from design education. In: Proceedings of CSCL-09 (Computer-Supported Collaborative Learning), Rhodes, Greece 227–231

8. Page, S.E.: The Difference: How the Power of Diversity Creates Better Groups, Firms, Schools, and Societies. Princeton University Press (2008)
9. Woolley, A.W., Chabris, C.F., Pentland, A., Hashmi, N., Malone, T.W.: Evidence for a collective intelligence factor in the performance of human groups. Science (New York, N.Y.) **330**(6004) (October 2010) 686–8
10. Barron, B.: When smart groups fail. The journal of the learning sciences **12**(3) (2003) 307–359
11. Dillenbourg, P., Schneider, D., Synteta, P.: Virtual Learning Environments. In Dimitracopoulou, A., ed.: 3rd Hellenic Conference "Information & Communication Technologies in Education", Rhodes, Greece (2002) 3–18
12. Daft, R., Lengel, R.: Organizational information requirements, media richness and structural design. Management Science **32**(5) (1986) 554–571
13. Goldman, R., Pea, R.D., Barron, B., Derry, S., eds.: Video research in the learning sciences. Lawrence Erlbaum Associates, Mahwah, NJ (2007)
14. Meier, A., Spada, H., Rummel, N.: A rating scheme for assessing the quality of computer-supported collaboration processes. International Journal of Computer-Supported Collaborative Learning **2**(1) (February 2007) 63–86

AIED 2013 Workshops Proceedings
Volume 2

Scaffolding in Open-Ended Learning Environments (OELEs)

Workshop Co-Chairs:

Gautam Biswas
Vanderbilt University, Nashville, TN. USA

Roger Azevedo
McGill University, Canada

Valerie Shute
Florida State University, FL. USA

Susan Bull
University of Birmingham, UK

https://sites.google.com/site/scaffoldingoeles/home

Preface

Open-ended learning environments (OELEs) offer students opportunities to take part in authentic and complex problem solving and inquiry tasks by providing a learning context and a set of tools for exploring, hypothesizing, and building their own solutions to problems. Also referred to as exploratory environments, examples include hypermedia learning environments, modeling and simulation environments, microworlds, scientific inquiry environments, and educational games featuring open worlds. OELEs may be characterized by choices students have as they are involved in their learning and problem solving tasks; in OELEs, students are faced with a multitude of decisions about what, when, and how to learn. Naturally, these choices offer critical opportunities for students to exercise higher-order skills that include:

- *Cognitive processes* for accessing, organizing, and interpreting information, constructing problem solutions, and assessing constructed solutions;
- *Metacognitive monitoring and self-regulatory processes* for coordinating the use of cognitive processes and reflecting on the outcome of solution assessments; and
- *Emotional and motivational self-regulatory processes* that include curiosity and persistence, especially in the face of difficulty.

This presents significant challenges to novice learners because they may not have the proficiency for using the system's tools, nor the experience and understanding necessary for explicitly monitoring and regulating their emotions and behaviours as they pursue learning goals. Not surprisingly, research has shown that novices often struggle to succeed in OELEs. Without *adaptive scaffolds*, these learners typically use tools incorrectly, adopt sub-optimal learning strategies for goal selection and planning, and fail to regulate key cognitive, motivational, and emotional processes. Adaptive scaffolds in OELEs refer to actions taken by the learning environment, based on the learner's interactions, intended to support the learner in completing a task and understanding the topic. Broadly, providing adaptive scaffolds consists of two sub-problems: (1) measuring and interpreting student behaviours to determine which adaptive scaffolds will be beneficial for their learning, and (2) providing adaptive scaffolds that effectively support student needs.

Given the developing interest in this area, this workshop sought papers on: (1) theoretical frameworks for designing scaffolding; (2) implementations of adaptive scaffolds; (3) cognitive, metacognitive and self-regulation models for designing scaffolds; and (4) formative assessments that support students' learning, performance, and learning-related behaviors. 14 papers have been accepted for this workshop: 8 as long papers that have each been allocated 8 pages, and 6 as short papers that have each been allocated 4 pages in the workshop proceedings.

A number of the accepted papers present games for learning science and math content as an open-ended learning environment where students have choice in constructing their own solutions to targeted problems. However, when the system detects non-optimal or incorrect behavior, it provides adaptive scaffolds to help the students discover and correct their incorrect solutions. Some of the papers discuss scaffolds in the form of representation schemes and selective tasks assigned to the student that aid their learning processes. Other papers use machine learning and data mining techniques to analyze student activity data and determine their learning behaviors and approaches to solving problems. A few papers adopt self-explanation as the framework for providing adaptive scaffolds, while others use Open Learner Modeling (OLM) as a mechanism for promoting student reflection, planning, and decision-making. One of the papers uses scaffolding to help students improve their metacognitive judgments. Another paper studies the effect of scaffolding as students work on invention activities related to data analysis. Finally, we also have a paper that discusses taxonomy of adaptive scaffolds in computer-based learning environments. We hope this set of papers leads to interesting and important discussions, and all participants can take away something that benefits their own work and advances the state of the art in this very important field of research.

In addition to the paper presentations and discussion, this workshop features other events:

1. A combined 90 minute hands-on activity and demonstration session where participants create levels to target and assess specific competencies in the Newton's Playground game (see http://www.gameassesslearn.org/newton/; the system has a level editor built into the game environment).
2. In the second half of the demonstration session, participants can demonstrate their creations.
3. A panel, where we compare and contrast approaches to scaffolding in traditional ITS problem solving environments and OELEs.

This workshop is the next in the series of Intelligent Support in Exploratory Environments (ISEE) Workshops that started in EC-TEL ’08 and has had representations in previous AIED, ITS and ICLS conferences. The last workshop was held at the Intelligent Tutoring Systems (ITS-2012) conference in Chania, Greece in June, 2012 (https://sites.google.com/a/lkl.ac.uk/isee/isee-its-12). Finally, we would like to acknowledge the contributions of all of the authors, without which this workshop would not have taken place. Many thanks to the program committee that helped review the submitted papers and provide valuable feedback to the authors. Last, but not the least, a special thanks to James Segedy, who helped put together the Workshop proceedings.

July 9, 2013
Gautam Biswas, Roger Azevedo, Valerie Shute, and Susan Bull

Program Committee

Workshop Co-Chairs
Gautam Biswas (Vanderbilt University): gautam.biswas@vanderbilt.edu
Roger Azevedo (McGill University): roger.azevedo@mcgill.ca
Valerie Shute (Florida State University): vshute@fsu.edu
Susan Bull (University of Birmingham): s.bull@bham.ac.uk

Workshop Committee

Vincent Aleven, Carnegie Mellon University

Bert Bredeweg, University of Amsterdam

Cristina Conati, University of British Columbia

Sergio Gutiérrez-Santos, London Knowledge Labs

Judy Kay, University of Sydney

Susanne Lajoie, McGill University

James Lester, North Carolina State University

Rose Luckin, London Knowledge Labs

Manolis Mavrikis, London Knowledge Labs

Bruce McLaren, Carnegie Mellon University

Ido Roll, University of British Columbia

James Segedy, Vanderbilt University

Philip Winne, Simon Fraser University

Table of Contents

Digital Games and Science Learning: Design Principles and Processes to Augment Commercial Game Design Conventions

Douglas B. Clark, Stephen Killingsworth, Mario Martinez-Garza, Grant Van Eaton, Gautam Biswas, John Kinnebrew, Pratim Sengupta, Kara Krinks, Deanne Adams, Haifeng Zhang, and James Hughes

Vanderbilt University
doug.clark@vanderbilt.edu, stephenkillingsworth@gmail.com, kwarizmi@gmail.com, grant.vaneaton@vanderbilt.edu, gautam.biswas@vanderbilt.edu, john.s.kinnebrew@vanderbilt.edu, pratim.sengupta@vanderbilt.edu, kara.krinks@gmail.com, deanne.adams@gmail.com, haifeng.zhang@vanderbilt.edu, jamesh53@gmail.com

Abstract. Digital games have the potential to make unique and powerful contributions to science education efforts. Much of that potential, however, remains unrealized, partly because powerful games for science learning need to synergistically augment commercial game design conventions and principles with design principles specific to the goals and nature of science learning and research on science learning. This paper builds on earlier frameworks outlining the affordances of commercial game design conventions for learning by proposing three design principles to help students explicitly articulate the intuitive science learning inherent in good game play in terms of formal science concepts and representations. We discuss these principles in the context of our recent and ongoing work in the SURGE projects. These projects investigate effective game mechanics to help students organize their tacit understandings about Newtonian mechanics into more formalized concepts.

Keywords: Digital learning environments, prediction, explanation, scaffolding, science education

1 Introduction

Digital games provide a promising medium for science education (Clark, Nelson, Sengupta, & D'Angelo, 2009; Honey & Hilton, 2010; NRC, 2009). In 2006, the Federation of American Scientists issued a widely publicized report stating their belief that games offer a powerful new tool to support education and encouraging private and governmental support for expanded research into complex gaming environments for learning. In 2009, a special issue of Science (Hines, Jasny, & Mervis, 2009) highlighted digital games in their survey of the promises and challenges of educational technology. Much of the initial debate over digital games for science education has focused on whether or not they support learning on science in general terms. This is obviously a simplistic question; well-designed games should produce better learning outcomes than games with unsound design. The NRC report on laboratory activities and simulations (Singer, Holton, & Schweingruber, 2005) supports this view, making clear that the design of physical and virtual learning activities, rather than simply the potential affordances of the medium, determines efficacy for learning. This paper outlines design principles focusing on helping students explicitly articulate the intuitive science learning inherent in good gameplay in terms of formal science concepts and representations.

2 SURGE I: Design and Rationale

SURGE was originally funded by an exploratory NSF DR-K12 grant between Vanderbilt University and Arizona State University (Clark & Nelson, 2008). The original design goal involved developing a game that would integrate formal physics representations and concepts with popular gameplay mechanics. We built SURGE I as a multi-platform game using the Unity3D game engine (unity3d.com). The SURGE I platform was intended to investigate design approaches for connecting students' “spontaneous concepts” (i.e., intuitions about kinematics and Newtonian mechanics) with formalized “instructed concepts.” The design approaches integrate (1) disciplinary representations of Newtonian mechanics and explicit connections to its central concepts with (2) popular commercial game mechanics from games such as Mario Galaxy and Switchball that include marble motion. As a result, SURGE I and SURGE II are conceptually-integrated games for learning (Clark & Martinez-Garza, in press), rather than conceptually-embedded games. The science to be learned is thus integrated directly into the mechanics of navigating through the game world, rather than being embedded as an activity to be visited at some location in the game environment. The latter structure is typically present in many virtual worlds designed for science learning.

We focused heavily on popular game-play mechanics from appropriate game genres in the design of SURGE I. Core ideas from commercial game design conventions included (a) supporting engagement and approachable entry (Koster, 2004; Squire, 2011), (b) situating the player with a principled stance and perspective (McGonigal, 2011), (c) providing context and identification for the player with a role and narrative (Pelletier, 2008; Aarseth, 2007; Gee, 2007;), (d) monitoring and providing actionable feedback for the player (Annetta et al., 2009; Garris, Ahlers & Driskell, 2002; Kuo, 2007; Munz, Schumm, Wiesebrock & Allgower, 2007), and (e) using pacing and gatekeeping to guide the player through cycles of performance (Squire, 2006). An extended review of these commercial game ideas would be outside the focus of this paper; they are discussed in full detail in the cited works and other excellent analyses of the affordances of commercial game design for learning (e.g., Annetta, 2010; Gee, 2009; Klopfer, Osterweil, & Salen, 2009).

3 Baseline Student Performance in Original Surge I Design

Students playing versions of SURGE I demonstrated high engagement and significant learning gains on items based on the highly-regarded Force Concept Inventory (FCI), which is a widely known benchmark assessment for conceptual understanding of Newtonian dynamics at the undergraduate level (Hestenes & Halloun, 1995; Hestenes, Wells, & Swackhamer, 1992). A study with 208 seventh and eighth grade students in Taiwan and 72 seventh grade students in the United States (Clark, Nelson, Chang, D'Angelo, Slack, & Martinez-Garza, 2011), for example, showed significant pre-post gains, t(250) = 2.0792, p (one-tailed)= 0.019, with modest effect sizes. In Taiwan, 62% of the students liked or really liked playing SURGE, 32% thought it was okay, and only 6% did not like it. In the United States, 76% of the students liked or really liked playing SURGE, 21% thought it was okay, and only 3% did not like it. These percentages were similar across gender and previous game-playing experience. These findings mirrored our findings in multiple studies conducted with different populations including: (a) 155 U.S. undergraduate physics students (D'Angelo, 2010), (b) 69 U.S. Title I sixth grade students, (c) 72 U.S. undergraduate educational psychology students (Slack et al. 2010), and (d) 124 U.S. undergraduate educational psychology students (Slack 2011). Those studies showed similarly significant pre-post gains (one-tailed p = .001, p = .02, p = .006, and p = .01, respectively).

The downside, however, was that these gains and increasing mastery focused on intuitive understanding (which is what the FCI largely measures) rather than explicit understanding. Essentially, players could more accurately predict the results of various actions, impulses, and interactions (which improves performance in the game and on FCI questions), but players were not being supported in explicitly articulating their mental models and the connections from choices made in game play to formal disciplinary representations and concepts.

Thus these results demonstrated that the players were developing intuitive rather than formal understandings while playing a game built mainly on commercial design principles. This makes sense because the goal of commercial games involves helping players develop robust intuitive understanding that helps them enjoy increasing levels of mastery as they play the game, which naturally increases their engagement and desire to play more. If players are left confused and unable to learn to play the game, or if the learning process is overwhelming or poorly structured, players will disengage, making it very unlikely that they will recommend the game to others or purchase future versions of the game. Repeated designs of this type would naturally drive a game company into bankruptcy. Thus, strong evolutionary pressures in the gaming industry favor design conventions that support intuitive understanding. There is no immediate market need, however, for commercial games to support explicit articulation or connection to formal ideas. The intuitive understandings developed at the heart of commercial games generally are not intended to correspond with important understandings outside of those games.

The use and purposes of the knowledge obtained from gameplay in commercial digital games diverge in some important respects from the goals for science education. Commercial game design conventions thus need to be augmented to meet the educational goals for science education. For learners to achieve the goals of science education, they must be supported in explicitly integrating the intuitive understanding they develop through popular game-play mechanics with formal disciplinary concepts and representations. This is a critical challenge for the design of games for science learning. How do we promote the integration of intuitive and formal learning without sacrificing the engaging intuitive learning encouraged by successful commercial gameplay?

Research in psychology, science education, and the learning sciences suggests a number of ways to support explicit articulation and integration, but the design principles developed through that research focus on contexts and mediums with different characteristics, affordances, and constraints than those of digital games. As result, in order to be synergistic rather than disruptive, these design principles from psychology, science education, and the learning sciences require adaptation and reinterpretation for the digital game medium. Two areas of research are of specific interest in our own work for leveraging explicit articulation in synergy with commercial game design conventions. These areas of research focus on enhancing (1) prediction within navigation interfaces, (2) self-explanation within game dialog.

4 SURGE II Design Approach: Prediction within Navigation Interfaces to Scaffold Model Articulation

Our SURGE II research explores the potential of leveraging the research on prediction and explanation from psychology and science education to engage students in reflecting more consciously and deliberately about the underlying physics models (e.g., Mazur, 1996; Grant, Johnson & Sanders, 1990; Scott, Asoko & Driver, 1991). Prediction and explanation can promote metacognition, learning, and reflection (e.g., Champagne, Klopfer & Gunstone, 1982) and conceptual change (Tao & Gunstone, 1999; Kearney, 2004; Kearney & Treagust, 2000). A growing body of research and scholarship on games and cognition emphasizes cycles of prediction, explanation, and refinement at the core of game-play processes (Salen & Zimmerman, 2004, Wright, 2006).

In terms of scaffolding prediction, SURGE II shifts mechanics to adapt to what we have learned from SURGE I. In SURGE II, players navigate their avatar through the play area to collect Fuzzies and treasures and deliver them to safe locations while avoiding obstacles and enemies (as in SURGE I). Rather than employing the real-time interfaces of the original SURGE grant (where pressing an “arrow key” resulted in immediate application of an impulse or constant thrust in the direction of the arrow key), the new versions incentivize prediction by requiring the player to spatially place all of the commands in advance. This feature has the advantage of requiring the player to make predictions about the results of each command in terms of the motion of the player’s avatar, rather than simply interacting reactively. Furthermore, SURGE II reduces the total number of commands a player initiates in a given level (thereby increasing the salience and impact of each individual command) to encourage players to think more carefully about the outcomes and implications of each action.

Our research with the new predictive interface to date has been promising. In our current study, 96 students played SURGE over three days. Learning outcomes were measured with an 11-item multiple-choice test of Newtonian kinematics modeled after the Force Concept Inventory and the Tennessee Comprehensive Assessment Program (TCAP) high-stakes science test. The pre- and post-test scores were compared using a two-sample paired t-test. The test showed a mean gain in test scores, from M = 3.48 to M = 4.51, and this result was statistically significant ($t = 5.184$, $p < .001$). The effect size was medium (Cohen's d = 0.57). Furthermore, the game was broadly appealing to students, with 92% of the respondents saying they "liked it" or "really liked it." Moreover, 80% of students considered the game appealing for both boys and girls. The sample comprised a cross-section of students who almost never play video games (40% reported playing less than two hours a week) as well as students for whom video games are a daily or near daily activity (33% reported playing an hour per day or more). These increased effect sizes encourage pushing forward with our exploration of leveraging prediction in the navigation interfaces.

5 SURGE II Design Approach: Self-Explanation within Game Dialog to Scaffold Model Articulation.

While the increased emphasis on prediction in the navigation design seems productive, the learning it promotes still focuses on making if/then predictions in the context of the consequences of different actions. We are, therefore, also exploring approaches for integrating explanation functionality into the dialog to leverage the increased intuitive grasp of the physics involved. Few games provide coherent structures for externalizing and reflecting on game-play; more often, such articulation and reflection occur outside the game, through discussion among players or participation in online forums (Gee, 2007; Squire, 2005; Steinkuehler & Duncan, 2008). We are now working to develop supports for this articulation and reflection by encouraging explanation and self-explanation in the dialog between the players and the characters within the game.

Research on self-explanation by Chi and others provides insight into the value of explanation for learning (e.g., Chi, Bassok, Lewis, Reimann, & Glaser 1989; Roy & Chi, 2005; Chi & VanLehn, in press). A recent review of research on students' self-explanation reports that self-explanation results in average learning gains of 22% for learning from text, 44% for learning from diagrams, and 20% for learning from multimedia presentations (Roy & Chi, 2005). Encouragingly, research by Bielaczyc et al. (1995) shows that instruction that stresses generating explanations improves performance even after the prompts that drive the explanations are discontinued. Mayer and Johnson (2010) have conducted preliminary work in embedding self-explanation in a game-like environment with encouraging results, including gains on transfer tasks. This emphasis on explanation is mirrored in research on science education. Work by White and Frederickson (1998, 2000), for example, demonstrates the value of asking students to reflect on their learning during inquiry with physics simulations.

Our design plan involves leveraging game dialog, which is a very popular aspect of conventional game design. Interestingly, while many aspects of commercial game design are currently very sophisticated, dialog in commercial games tends to involve relatively simple "multiple-choice" dialog trees that are not difficult to create. In fact, dialog in games is an area where educational games could take the lead. In SURGE II, after a player has completed a set of missions in the core game, a computer-controlled character in the game contacts the player and asks for help in mounting a similar rescue mission. The plan is for the resulting dialog tree to scaffold the player, requiring him or her to construct a solution for the character and to convince the character to try the solution by explaining how it fits a larger pattern of phenomena related to Newton's three laws of motion. Our goal is to present these invitations for dialog as puzzles that are engaging in their own right (Clark & Martinez-Garza, in press; Clark, Martinez-Garza, Biswas, Luecht, & Sengupta, in press). We will conduct our first studies of this approach later this year and will continue to explore its affordances for explicit articulation.

6 Bibliography

1. Aarseth, E. (2007). I Fought the Law: Transgressive Play and The Implied Player. Proceedings of DiGRA 2005 Conference: Situated Play.
2. Annetta, L. A. (2010). The "I's" have it: A framework for serious educational game design. Review of General Psychology, 14(2), 105.
3. Annetta, L. A., Minogue, J., Holmes, S. Y., & Cheng, M.-T. (2009). Investigating the impact of video games on high school students' engagement and learning about genetics. Computers and Education, 53(1), 74-85.
4. Bielaczyc, K., Pirolli, P., & Brown, A. L. (1995). Training in self-explanation and self-regulation strategies: Investigating the effects of knowledge acquisition activities on problem solving. Cognition and Instruction, 13(2), 221-252.
5. Champagne, A. B., Klopfer, L. E., & Gunstone, R. F. (1982). Cognitive research and the design of science instruction. Educational Psychologist, 17(1), 31.
6. Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. Cognitive Science, 13(2), 145–182.
7. Chi, M. T. H., & VanLehn, K. A. (in press). The content of physics self-explanations. Journal ofthe Learning Sciences.
8. Clark, D. B., & Nelson, B. (2008). Scaffolding Understanding by Redesigning Games for Education (SURGE). Exploratory DR-K12 grant funded by the U.S. National Science Foundation, 2008-2012
9. Clark, D. B., Nelson, B., Sengupta, P., D'Angelo, C. M. (2009). Rethinking Science Learning Through Digital Games and Simulations: Genres, Examples, and Evidence. Paper commissioned for the National Research Council Workshop on Games and Simulations. Washington, D.C.
10. Clark, D. B., Nelson, B. C., Chang, H.-Y., Martinez-Garza, M., Slack, K., & D'Angelo, C. M. (2011). Exploring Newtonian mechanics in a conceptually-integrated digital game: Comparison of learning and affective outcomes for students in Taiwan and the United States. Computers & Education, 57(3), 2178-2195. doi:16/j.compedu.2011.05.007

11. Clark, D. B., & Martinez-Garza, M. (in press). Prediction and explanation as design mechanics in conceptually-integrated digital games to help players articulate the tacit understandings they build through gameplay. C. Steinkuhler, K. Squire, & S. Barab (Eds.), Games, learning, and society: Learning and meaning in the digital age. Cambridge: Cambridge University Press.
12. Clark, D. B., Martinez-Garza, M., Biswas, G., Luecht, R. M., & Sengupta, P. (in press). Driving Assessment Of Students' Explanations in Game Dialog Using Computer-Adaptive Testing and Hidden Markov Modeling. In D. Ifenthaler, D. Eseryel, & G. Xun (Eds.). Game-based Learning: Foundations, Innovations, and Perspectives. New York: Springer.
13. D'Angelo, C.M. (2010). Scaffolding vector representations for student learning inside a physics game. Unpublished doctoral dissertation. Arizona State University.
14. Garris, R., Ahlers, R., & Driskell, J. E. (2002). Games, Motivation, and Learning: A Research and Practice Model. Simulation & Gaming, 33(4), 441 -467. doi:10.1177/1046878102238607
15. Gee, J. P. (2007). What Video Games Have to Teach Us About Learning and Literacy. Second Edition: Revised and Updated Edition (2nd ed.). Palgrave Macmillan.
16. Gee, J. P. (2009). Deep learning properties of good digital games: How far can they go. Serious Games: Mechanisms and Effects. Taylor & Francis Group, Routledge.
17. Grant, P., Johnson, L., Sanders, Y., & Science Teachers' Association of Victoria. (1990). Better links : teaching strategies in the science classroom. Melbourne: Science Teachers' Association of Victoria.
18. Hestenes, D., & Halloun, I. (1995). Interpreting the Force Concept Inventory: A Response to March 1995 Critique by Huffman and Heller. Physics Teacher, 33(8), 502–506.
19. Hestenes, D., Wells, M., & Swackhamer, G. (1992). Force Concept Inventory. Physics Teacher, 30(3), 141–158.
20. Hines, P. J., Jasny, B. R., & Merris, J. (2009). Adding a T to the three R's. Science, 323, 53.
21. Honey, M. A., & Hilton, M. (Eds.). (2010). Learning Science Through Computer Games and Simulations. National Research Council. Washington, DC: National Academy Press.
22. Kearney, M. (2004). Classroom Use of Multimedia-Supported Predict–Observe–Explain Tasks in a Social Constructivist Learning Environment. Research in Science Education, 34(4), 427–453.
23. Kearney, M., & Treagust, D. F. (2000). An investigation of the classroom use of prediction-observation-explanation computer tasks designed to elicit and promote discussion of students' conceptions of force and motion. Presented at the National Association for Research in Science Teaching, New Orleans, USA.
24. Klopfer, E., Osterweil, S., & Salen, K. (2009). Moving Learning Games Forward. Cambridge, MA: The Education Arcade.
25. Koster, R. (2004). A Theory of Fun for Game Design (1st ed.). Paraglyph Press
26. Kuo, M.-J. (2007). How does an online game based learning environment promote students' intrinsic motivation for learning natural science and how does it affect their learning outcomes? Digital Game and Intelligent Toy Enhanced Learning, 2007. DIGITEL '07. The First IEEE International Workshop on (pp. 135-142). Presented at the Digital Game and Intelligent Toy Enhanced Learning, 2007. DIGITEL '07. The First IEEE International Workshop on. doi:10.1109/DIGITEL.2007.28
27. Mayer, R. E., & Johnson, C. I. (2010). Adding instructional features that promote learning in a game-like environment. Journal of Educational Computing Research, 42(3), 241–265.
28. Mazur, E. (1996). Peer Instruction: A User's Manual (Pap/Dskt.). Benjamin Cummings.

29. McGonigal, J. (2011). Reality Is Broken: Why Games Make Us Better and How They Can Change the World. New York, NY: Penguin Press.
30. Munz, U., Schumm, P., Wiesebrock, A., & Allgower, F. (2007). Motivation and Learning Progress Through Educational Games. Industrial Electronics, IEEE Transactions on, 54(6), 3141-3144. doi:10.1109/TIE.2007.907030
31. National Research Council. (2009). National Research Council Workshop on Games and Simulations. October 6-7, 2009, Washington, D.C.
32. Pelletier, C. (2008). Gaming in Context: How Young People Construct Their Gendered Identities in Playing and Making Games. In Y. B. Kafai, C. Heeter, J. Denner, & J. Y. Sun (Eds.), Beyond Barbie and Mortal Kombat: new perspectives on gender and gaming. Cambridge, Mass: The MIT Press.
33. Roy, M., & Chi, M. T. H. (2005). The self-explanation principle in multimedia learning. In R. E. Mayer (Ed.), The Cambridge handbook of multimedia learning (pp. 271-286). New York: Cambridge University Press.
34. Salen, K., & Zimmerman, E. (2004). Rules of play: game design fundamentals. 2004.
35. Scott, P. H., Asoko, H. M., & Driver, R. H. (1991). Teaching for conceptual change: A review of strategies. Connecting Research in Physics Education with Teacher Education, 71–78.
36. Singer, S. Hilton, M.L., & Schweingruber, H.A. (Eds.) (2005). America’s lab report: investigations in high school science. Washington, DC: National Academies Press.
37. Slack, K., Nelson, B., Clark, D. B., & Martinez-Garza, M. (2011). Model-Based Thinking in the Scaffolding Understanding by Redesigning Games for Education (SURGE) Project. Poster presented as part of a structured poster session at the American Educational Research Association (AERA) 2011 meeting. New Orleans, LA.
38. Slack, K. Nelson, B., Clark, D. B., Martinez-Garza, M. (2010). Influence of visual cues on learning and in-game performance in an educational physics game environment. Paper presented at the Association for Educational Communications and Technology (AECT) 2010 meeting. Anaheim, California.
39. Squire, K. (2005). Changing the game: What happens when video games enter the classroom. Innovate: journal of online education, 1(6), 25–49.
40. Squire, K. (2006). From content to context: Videogames as designed experience. Educational Researcher, 35(8), 19.
41. Squire, K. (2011). Video Games and Learning: Teaching and Participating Culture in the Digital Age. New York: Teachers College Press.
42. Squire, K. D., DeVane, B., & Durga, S. (2008). Designing centers of expertise for academic learning through video games. Theory Into Practice, 47(3), 240–251.
43. Steinkuehler, D., & Duncan, S. (2008). Scientific habits of mind in virtual worlds. Journal of Science Education and Technology, 17(6), 530-543.
44. Tao, P.-K., & Gunstone, R. F. (1999). The process of conceptual change in force and motion during computer-supported physics instruction. Journal of Research in Science Teaching, 36(7), 859–882.
45. White, B., & Frederiksen, J. (1998). Inquiry, modeling, and metacognition: Making science accessible to all students. Cognition and Instruction, 16(1), 3-118.
46. White, B. Y., & Frederiksen, J. R. (2000). Metacognitive facilitation: An approach to making scientific inquiry accessible to all. In J. A. Minstrell & E. H. Van Zee (Eds.), Inquiring into Inquiry Learning and Teaching in Science (pp. 331-370). American Association for the Advancement of Science.
47. Wright, W. (2006). Dream machines. Wired 14(04).

Understanding Users' Interaction Behavior with an Intelligent Educational Game: Prime Climb

Alireza Davoodi[1], Samad Kardan[1], Cristina Conati[1]
[1] Computer Science Department, The University of British Columbia,
ICICS/CS Building, 201-2366 Main Mall, Vancouver, B.C., Canada V6T 1Z4,
{davoodi, skardan, conati}@cs.ubc.ca

Abstract. This paper presents work on applying clustering and association rule mining techniques to mine users' behavior in interacting with an intelligent educational game, Prime Climb. Through such behavior discovery, frequent patterns of interaction which characterize different groups of students with similar interaction styles are identified. The relation between the extracted patterns and the average domain knowledge of students in each group is investigated. The results show that the students with significantly higher prior knowledge about the domain behave differently from those with lower prior knowledge as they play the game and that pattern could be identified early during the interactions.

Keywords: Intelligent Educational Games, Behavior Discovery, Association Rule Mining, Open Ended Learning, Scaffolding

1 Introduction

Open-Ended Learning Environments (OELEs) support student-centered learning and allow learners to follow an exploratory interaction behavior to construct their own models of concepts and revise their beliefs subsequent to receiving immediate feedback on their actions [1]. Previous studies have shown that students could not benefit much from an open-ended learning environment if not receiving proper feedback [2]. Among learning environments, educational games are designed to foster motivation and engagement which are shown to be influential in learning [3]. To this end, educational games such as Crystal Island provide exploratory learning environments and encourage autonomous interaction with the game [4]. While such freedom in interaction is required to maintain engagement in the game, it also provides learners with the possibility of showing different interaction patterns. The interaction patterns might be indicative of certain characteristics and understanding such patterns can provide valuable information about the students.

Adaptive OELEs have been designed to answer the need for understanding and intelligently supporting varying learning styles, capabilities, and preferences in individuals in developing their skills. An adaptive educational system maintains a model of student's learning and leverages the student's interactions with the system to provide tailored scaffolding. Many educational systems apply data mining approaches on the logs of students' recorded interactions to extract behavioral patterns and extract high-level information about students [5-7]. Along this line of research, we concentrate on understanding how students interact with Prime Climb (PC), an adaptive educational game (edu-game) and whether there is a connection between behavior patterns of students and their attributes such as prior knowledge. The ultimate goal in an adaptive educational game such as PC is to help a higher number of students learn the desired skills through interacting with the game. Achieving such an objective requires a pedagogical agent which maintains an accurate understanding of individual differences among users and provides more tailored interventions, with the aim of guiding the learners in the right learning direction. For instance, if a pedagogical agent is capable of identifying a group of students with high domain knowledge, it is possible to leverage such information to construct a more accurate user model and intervention mechanism. The user's interaction behaviours can also be provided to developers to improve the design of educational systems [8].

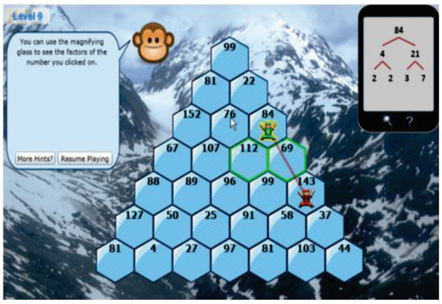
Behavioral discovery has been vastly used in educational systems, but there is limited application in educational games such as Prime Climb, in which educational concepts are embedded and presented in the game scenarios and narratives with minimum explicit technical notation (for instance mathematical notations in PC) to more genuinely support game aspects of the system. In Prime Climb, students do not explicitly practice approaches to number factorization but implicitly follow a self-regulated learning approach [9] to explore and understand the methods and practice them. This paper describes the first step toward leveraging students' behavioral patterns into building a more effective adaptive edu-game. The ultimate goal is devising mechanisms for extracting abstract high-level patterns from raw interaction data and leveraging such understanding for real-time identification of interaction styles to enhance user modeling and intervention mechanism in an edu-game like PC.

Behavior discovery has been recently applied in different educational systems. Kardan et al. [6] leveraged behavior discovery to propose a general framework for distinguishing users' interaction styles in exploratory learning environments. Keshtkar et al. [10] describe an approach to distinguishing players and mentors roles in a multi-chat environment within the epistemic game Urban Science. In another related work, Mccuaig et al. [5] discuss using interaction behaviors to distinguish students who will fail or pass a course in a Learning Management System (LMS). A sequence mining approach has been also used in differentiating behavior patterns in students' interacting with Betty's Brain, a learning-by-teaching environment [7].

Although behavior discovery has been recently applied to many educational systems, there is very limited work on behavior mining in an open ended intelligent educational game like Prime Climb in which learning through playing the game is intended. Additionally, most of the previous works use the entire interaction data to make inferences about the users. In this work, we present the results of behavior mining not only on a big portion of interaction data but also on a truncated data set, which will provide the possibility of constructing an online classifier for early detection of varying patterns of interactions.

2 Prime Climb an Intelligent Edu-game

Prime Climb (PC) is an intelligent educational game for students in grades 5 and 6 to practice number factorization skills. Prime Climb is equipped with an intelligent pedagogical agent which maintains a probabilistic model of the student's knowledge on number factorization skills.



You can use the magnifying glass to see the factors of the number you clicked on.

Fig. 1 Prime Climb

The pedagogical agent leverages the probabilistic model to provide an adaptive scaffolding mechanism. If model's assessment about the student's knowledge on a skill falls below a certain threshold, a hint is presented to the player. The hints are given in incremental level of details. In PC, the player and his/her partner climb a series of 11 mountains of numbers by pairing up the numbers which do not share a common factor. There are two main interactions of a player with PC:

Making Movements**:** A player makes one or more movements at each time, by clicking on numbered hexagons on the mountains. PC provides immediate feedbacks on correctness of movements. If a player makes a wrong movement, s/he falls down.

Using Magnifying Glass Tool**:** The magnifying glass (MG) tool is always available for the user to benefit from. The MG is used to show the factor tree of a number on the mountains; it is located in the top right corner of the game (Fig. 1).

4 Behavior Discovery in Prime Climb

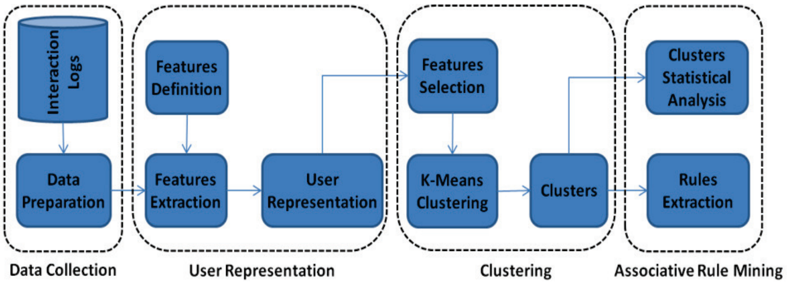


Fig. 2: Behavior discovery methodology in Prime Climb

4.1 Data Collection

Data collection is first component of the behavior discovery methodology in PC shown in Fig. 2. We collected interaction logs of 45 students who played PC voluntarily. Prime Climb consists of 11 levels (mountains), and not all students could manage to reach the last level. Out of the 45 students, 43 completed 9 or more levels. The remaining 2 students who completed fewer levels were excluded from further analyses to ensure that all students in analysis had completed a minimum of 9 levels. For the remaining 43 students, the interaction data for the first 9 mountains was used in the feature extraction process.

4.2 User Representation

Features Definition: Each user is represented by a vector of features. Based on the 2 main groups of interaction previously mentioned (movements and MG), two types of features are defined: (1) Movements features based on statistical measures on movements students made on the mountains and (2) MG features: based on statistical measures on students' usages of the MG tool. Table 1 shows some of these features:

Table 1 Some features used for behavior discovery

Movement Features
[Sum/Mean/STD] of number of [correct/wrong] movements made by a student across mountains
[Sum/Mean/STD] of time on [correct/wrong] movements made by a student across mountains
[Mean/STD] of length of sequences of [correct/wrong] moves made by a student
[Mean/STD] of time spent per sequence of [correct/wrong] moves made by a student
Magnifying Glass (MG) Features
[Sum/Mean/STD] of MG usage
Mean number of [correct/wrong] movements per each MG usage
STD of number of [correct/wrong] movements per each MG usage

Feature Set Definition: Each feature is a measure computed based on user's interactions with one or more mountains. There are two types of feature:

Mountain-Generic Features (m – n), (m >= 1 and n <= 9)**:** Calculated based on the users' interactions with mountains *m* to *n*, inclusively. For instance, the feature, correct-movements (1–9), represents the total number of correct movements made by the user on mountains 1 to 9.

Mountain-Specific Features (k), (1 <= k <= 9)**:** Calculated based on interactions with mountain *k*. For instance, correct-movements (7), represents the total number of correct movements made by the user on mountain 7.

In this paper, we present the behavior discovery results on the two feature sets:

Mountain-Generic Movement(1–9) Set**:** Contains mountain-generic features (1–9) which are related to movement actions the student makes.

Mountains-Generic+Specific-MG+Movement(1–2) Set**:** Contains mountain-generic MG features (1–2), mountain-generic movement features (1–2), mountain-specific MG features (1) and (2), and mountain-specific movements features(1) and (2).

4.3 Clustering

Feature Selection: Prior to performing clustering, feature selection is applied to filter out irrelevant features [11].

Clustering: The optimal number of clusters is determined as the lowest number suggested by C-index, Calinski and Harabasz[12] and Silhouette [13] measures of clustering validity. Once all the students are represented by vectors of selected features, the GA *K*-means (*K*-means for short) clustering algorithm [6], which is a modified version of GA *K*-means [14], is applied to cluster the users into an optimal number of clusters.

4.4 Rule Mining: Higher Prior Knowledge *vs.* Lower Prior Knowledge

Next, the Hotspot algorithm [15] is used to extract the rules for each discovered cluster. Also, we analyzed whether the resulting clusters are significantly different on a measure called *cluster's prior knowledge*, which is defined as follows:

Cluster's Prior knowledge: The cluster's prior knowledge gives the average level of factorization skills of the cluster's members prior to playing the game and is defined as the average of raw pre-test scores of the cluster's members. The following formula is used to calculate the cluster's prior knowledge:

$$\mathbf{Cluster's\ prior\ knowledge} = \frac{\sum_{\mathbf{student} \in \mathbf{cluster}} \mathbf{pre\_test(student)}}{\mathbf{Cluster's\ size}}. \tag{1}$$

where $\mathrm{pre\_test(student)}$ is the student's pre-test score. Before playing the game, a student takes a pre-test on number factorization skills. The maximum score a student can get is 15. The average pre-test score across the 43 students is 11.7, and the standard deviation is 3.29.

Behavior Discovery on Mountain-Generic-Movement(1–9) set: In this feature set, each student is represented by a vector of mountain-generic movement features(1–9). As a result of the features' selection mechanism, 18 features were selected out of the original 30 features. The optimal number of clusters was found to be 2, and the *K*-means method was used to cluster the set of students into 2 groups. The result of a *t*-test showed that there is a statistically significant difference between the prior knowledge of cluster 1 of students (higher prior knowledge (HPK) group) (*M* = 13.0 , *SD* = 2.0) and cluster 2 of students (lower prior knowledge (LPK) group) (*M* = 11.3 , *SD* = 3.45), *p*=.03 and Cohen's *d*= 0.53. Next, the Hotspot association rule mining algorithm was applied on the clusters to extract the associative rules. Table 2 shows the rules extracted for each cluster.

Understanding the Rule Mining Results:

Rules: Each bulleted item in following tables shows an extracted rule. For example, "Mean-Time-on-Movements=Higher" is an extracted rule which applies to at least 25% of the members of cluster 1. (In this study, the threshold of 25% is applied for all rules extracted by the Hotspot algorithm). This rule shows that the values for the feature "Mean-Time-On-Movements(1–9)" across the cluster's members belong to the "Higher" Bin.

Bins: In this study, the Hotspot algorithm considers two bins for values of each feature: (1) Lower bin and (2) Higher bin. Each bin shows a range of values of the features such that the lower bin represents the lower range of values and the higher bin represents the upper range of values for the feature. The cut-off point for splitting a range of values for a feature into two ranges (lower and upper) is calculated specifically for the feature in each extracted rule by the Hotspot algorithm. The lower and higher bins are indicated by the words "Lower" and "Higher" in front of the features in the following tables.

Rule's Support: The other important information is the rule's support shown in square brackets in front of the extracted rules in the following tables. For instance, [6/6=100%] in front of the first rule for the cluster 1 in Table 2 shows that there are in total 6 (in denominator) out of 43 students on which the extracted rule applies and all of these students belong to cluster 1 (6 in the numerator of the fraction). In addition, it can be concluded that this extracted rule applies to 60% (6/10) of the cluster 1 (note that the size of cluster 1 is 10).

Table 2: Extracted Rules for Mountains-Generic-Movement(1–9)

Rules for Cluster 1[HPK]: (Size: 10/43 = 23.26%)
• Mean-Time-on-Movements(1-9) = Higher, [6/6=100%] • Mean-Time-Spent-On-Correct-Movements-On-Mountains(1-9) = Higher, ([5/5=100%])
Rules for Cluster 2[LPK]: (Size: 33/43 = 76.74%)
• Mean-Time-On-Movements(1-9) = Lower, [33/37=89.19%] o STD-Time-On-Wrong-Correct-Moves(1-9) = Lower, [33/35=94.29%] • Mean-Time-On-Consecutive-Wrong-Movements(1-9) = Lower, [31/35=88.57%] o STD-Time-On-Movements(1-9) = Lower, [31/33=93.94%] o STD-Time-On-Correct-Movements(1-9) = Lower, [31/33=93.94]

Discussion and Interpretation: The extracted rules show that the students belonging to the HPK cluster (cluster 1) spent more time on movements and correct movements across 9 mountains. This could indicate that the students with higher prior knowledge were more involved in the game and spent more time before making a movement. Since the time spent on making a correct movement is higher for this group of students, it might mean that a correct move by this group of students is less likely to be due to a lucky guess as compared with the total population. In contrast, the group of students with lower prior knowledge spent less time on making movements as well as making wrong movements. This could be an indication of less involvement in the game by the lower prior knowledge group. It could show that a correct movement by this group of students is more likely due to a guess. In addition, there are some other frequent patterns of interaction for the group of students with lower prior knowledge. These patterns show a lower standard deviation on time spent on making movements and correct movements. This indicates that this group of students showed a consistent pattern of lack of engagement in the game. Therefore, we can conclude that the students with higher prior knowledge showed more engagement in the game than students with lower prior knowledge.

Behavior Discovery on Mountain-Generic+Specific-MG+Movement(1–2) set: This feature set does not employ interaction data from all 9 mountains; instead, only the data from the first 2 mountains is included. Such feature set is mainly valuable for constructing an online classifier to classify students based on their interaction with the game during the game play. The ultimate aim is leveraging such a feature set to step toward building a more accurate individualized student model and intervention mechanism as the student makes progress in the game. For instance, if the classifier can identify a student as a lower/higher knowledgeable student, it could leverage the information for early adjustment of the adaptive intervention mechanism. Similarly *K*-means was applied to cluster the students represented by the Mountains-Generic+Specific-MG+ Movements (1–2) Set. The optimal number of clusters was calculated to be 2, and 25 features out of 51 original ones were selected, as a result of applying the features selection mechanism. The cluster's *prior knowledge* was calculated for each of the discovered clusters and compared using a *t*-test. The result of the *t*-test showed a statistically significant difference between cluster *1's prior knowledge* (M = 12.45 , SD = 2.66) and cluster 2's prior knowledge (M = 9.22 , SD = 3.93), p = 0.02, Cohen's d = 1.08. Next, association rule mining was applied on the 2 clusters, as shown in Table 3.

***Interpretation and Discussion*:** As shown in Table 3, the results of behavioral discovery on the Mountains-Generic+Specific-MG+Movements(1-2) set is not consistent with the results of behavior discovery on the Mountains-Generic-Movements(1-9) set. Behavior discovery on interaction data from the first two mountains shows that students with higher prior knowledge (*M* = 12.45 , *SD* = 2.66) constitute around 79% of the all students and spend less time on making movements. It was previously shown in Table 2 that the students in the HPK cluster constituted approximately 23% of all students and spent more time on making movements when interaction data from all 9 mountains was included. Despite this, we expect that as the students progress in the game, the students with higher prior knowledge would behave differently from the other students and separate themselves from the others. To verify this, we also extracted frequent patterns when more interaction data from upper mountains is included in the clustering and rule mining. When the interaction data from the first 3 mountains is included in patterns mining, 2 clusters are identified which are not significantly different on their prior knowledge. When interaction data from the first four mountains is included, we observe patterns similar to those identified using the interaction data from all 9 mountains as shown in Table 3-right. The result of the *t*-test shows a statistically significant difference between cluster 1's prior knowledge (*M* = 13.28, *SD* = 1.58) and cluster 2's prior knowledge (*M* = 11.39 , *SD* = 3.4), *p* = 0.02, Cohen's *d* = 0.60. Also, approximately 16% of students belong to the HPK cluster, and 84% belong to the LPK group. This result is very similar to the results when data from all 9 mountains is included. Similar patterns are observed when more interaction data from upper mountains is included in the analysis.

Table 3: Extracted Rules for Mountains-Generic+Specific-MG+ Movements(1-2) [left] and MG+Movements(1-4) [right]

Rules for Cluster 1[HPK] **(Size: 33/42=78.57%)**	**Rules for Cluster 1[HPK]** **(Size: 7/43=16.28%)**
• Mean-Time-On-Movements(1)=Lower, [30/31 =96.77%] • Mean-Time-On-Movements(1-2) = Lower, [29/30 = 96.67%]	• Mean-Time-On-Movements(4) = Higher, [5/5 = 100%] • Mean-Time-On-Correct-Movements(3) = Higher, [3/3 = 100%]
Rules for Cluster 2[LPK] **(Size: 9/42=21.43%)**	**Rules for Cluster 2[LPK]** **(Size: 36/43=83.72%)**
• Mean-Time-Spent-On-Mountain(1-2) = Higher, [7/7=100%] • Total-Time-On-Mountain(1) = Higher, [5/5=100%]	• Mean-Time-On-Correct-Movements(1-4) = Lower, [35/35 = 100%] • Mean-Time-On-Movements(1-4)=Lower, [34/34 = 100%]

5 Conclusions and Next Steps

This paper discusses behavior discovery in Prime Climb (PC). To this end, different sets of features were defined. The features were extracted from interaction of students with PC in the form of making movements from one numbered hexagon to another numbered hexagon and usages of the MG tool. To identify frequent patterns of interaction, first, a feature selection mechanism was applied to select more relevant features from the set of all features. Then a *K*-means clustering was applied to cluster the students into an optimal number of clusters and the Hotspot algorithm of association rule mining was applied on the clusters to extract frequent interaction patterns. Finally, the prior knowledge of the clusters were compared. When interaction data from all 9 mountains was included in behavior discovery, it was found that the students with higher prior knowledge were more engaged in the game and spent more time on making movements. In contrast, the students with lower prior knowledge spent less time on making movements, indicating that they were less involved in the game. Behavior discovery also was conducted on truncated sets of features in which only a fraction of interaction data was included. The results showed that using the interaction data from the first two mountains resulted in groups of students that are statistically different on their prior knowledge.

The scaffolding mechanism in PC relies on the student model so we expect improvements in the model to result in more tailored interventions and guidance. Current PC uses the same student model for all students. Following the results of the presented study, we plan to adjust the model based on the characteristics of each discovered group of students. In addition, an online classifier will be built which identifies frequent patterns of interaction in the students, classifies them into different groups in real time, and leverages such information to build a more personalized user model and adaptive intervention mechanism in PC.

References

1. Papert, S. Mindstorms (2nd ed.), New York: Basic Books inc. (1993) _
2. L. Alfieri, P. Brooks, N. Aldrich, and H. Tenenbaum, “Does Discovery-Based Instruction Enhance Learning,” Journal of Education Psychology, vol. 103, no. 1, pp. 1-18, (2011)
3. Lee, J., Luchini, K., Michael, B., Norris, C., Solloway, E., 2004, More than just fun and games: Assessing the value of educational video games in the classroom. Proceedings of ACM SIGCHI 2004, Vienna, Austria, pp. 1375–1378 (2004)
4. Sabourin, J., Rowe, J., Mott, B., & Lester, J. Exploring Affect and Inquiry in Open-Ended Game-based Learning Environments. (2011)
5. Mccuaig J., Baldwin, J.: Identifying Successful Learners from Interaction Behaviour. EDM 2012, Chania, Greece. pp. 160–163 (2012).
6. Kardan, S., and Conati. C.: A Framework for Capturing Distinguishing User Interaction Behaviours in Novel Interfaces. EDM, Eindhoven, the Netherlands, pp. 159–168 (2011)
7. Kinnebrew, J. S., Biswas G.: Identifying Learning Behaviors by Contextualizing Differential Sequence Mining with Action Features and Performance Evolution, EDM 2012, Chania, Greece. pp. 57–64 (2012).
8. Hunt, E., Madhyastha, T.: 2005: Data Mining Patterns of Thought. In: Proceedings of the AAAI Workshop on Educational Data Mining. (2005).
9. Paris, S., Paris, A.: Classroom Applications of Research on Self-Regulated Learning. Educ. Psychol. 36(2), 89–101 (2001)
10. Keshtkar, F., Morgan, B., Graesser, A.: Automated Detection of Mentors and Players in an Educational Game. EDM 2012, Chania, Greece. pp. 212–213 (2012)
11. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. 3, 1157–1182 (2003).
12. Milligan, G.W., Cooper, M.C.: An Examination of Procedures for Determining the Number of Clusters in a Data Set. Psychometrika, 50(2), 159–179 (1985).
13. Rousseeuw, P. J. :Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis". *Computational and Applied Mathematics* 20: 53–65. doi:10.1016/0377-0427(87)90125-7, (1987)
14. Kim, K. Ahn, H.: A Recommender System using GA *K*-means Clustering in an Online Shopping Market. Expert Syst. Appl. 34(2), 1200–1209 (2008).
15. Hall, M., Eibi, F., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data mining Software: An Update. SIGKDD Explor. 11(1), 10–18 (2009)

Designing Digital Objects to Scaffold Learning

Grant Van Eaton, Douglas B. Clark, David Beutel

Vanderbilt University, Peabody College
grant.vaneaton@vanderbilt.edu, doug.clark@vanderbilt.edu,
dmbeutel@gmail.com

Abstract. Digital objects in learning games provide opportunities to scaffold teacher and student learning toward deeper epistemological understanding of the concepts they represent. Representations encapsulated in digital objects, however, have the potential to misrepresent the concepts they stand in place of. Using student and teacher interview data after playing a physics learning game, analysis of the role of representations in students' epistemological development led to two design recommendations. When designing digital objects to effectively scaffold concepts, designers should pay attention to the ways in which learning environments explore the nature of core concepts represented by digital objects and explicitly model the meaning of the representations in the learning environment.

Keywords: Digital learning environments, representation, scaffolding, epistemology, science education

1 Introduction

In their review of the literature on digital games and simulations for science education, Clark, et al [1] propose a shift in research agenda away from an exploratory phase that furnishes mere proofs of concept and instead calls on researchers to focus on ascertaining the design principles that best support learning and conceptual change. Design principles in digital learning environments necessarily rely on the use of representations that interact with players in order to model core concepts. These representations then have the power to scaffold the learning trajectory of both teachers and students as they play the game. Representations, however, have the ability to take on a life of their own as a teacher or student appropriates them as tools for learning. Using interview data collected from a four-day classroom implementation of the SURGE: EPIGAME physics learning game, this paper will explore two questions central to the interplay between design, representation, and epistemology:

- How do representations in the SURGE learning environment interact with teachers and students?
- How do these representations scaffold the development of teachers' and students' epistemology of force?

1.1 Theoretical Framework

When thinking about how to use representations to scaffold concepts in a digital learning environment, Ball and Cohen's [2] educative curriculum framework provides an orientation that positions the learning environment to scaffold learning not only for students, but also their teachers. Using learning games to develop deeper content knowledge in teachers, however, will only be effective insofar as 1) the representations in the learning environment properly embody the focus concept(s) and 2) if the correct scaffolds are in place to bridge teachers' intuitive understanding of their content with the concepts represented in the game.

1.2 Representations in the Learning Environment

In order to discuss the potential for learning games to educate students, and the importance of representations to accomplish this task, this analysis will focus on a key representation in the SURGE: EPIGAME learning environment: force. In SURGE, players must navigate a spaceship around obstacles while staying on a set path. This is accomplished by issuing commands to the ship as to the magnitude and direction the ship should fire forces to achieve the desired path. Within the game, these representations are represented by force tiles placed on a timeline delineated in one-second increments.

As representations in the game, force tiles are intended to represent a command given to the ship to fire a force of a specific magnitude and direction at a certain time. This representation is not the actual force being applied, but rather a command to the ship to fire the desired force. Force tiles are placed within the timeline at the bottom of the simulation space, representing when the ship should issue the command to fire the force indicated on the force tile. The timeline is thus intended to represent and visualize the amount of time between commands to fire forces.

2 Impact of Representations on Scaffolding Learning

Lehrer and Schauble [3] have shown that representations edit concepts insofar as they reduce or enhance the information they contain. In the best case scenario, these reductions and enhancements effectively scaffold student and teacher understanding toward the concept embodied in the representation. These representations, however, also have the potential to misrepresent the concept to such an extent that, despite the best design intentions, students and teachers emerge from interaction with the representation holding a fundamentally different concept than intended by designer.

2.1 Force

Throughout student interviews, force tiles take on independent ontological status as actors in the game's simulation space, contrary to the intent of the designers. One student repeatedly talks of 'sending' a force from the timeline into the simulation space in order to do work, even gesturing from force tiles in the timeline to the point in the simulation space in which the force is applied:

Student: Like, where it sends... where you send a 60 Newton force over here to get to this point, and then you'd send another 60 Newton force to stop it *[student gestures from 60 Newton force on timeline to the spot where the force is applied in the simulation space]* ... and then a 20 Newton force... *[repeats gesture]* and then a 20 Newton force to stop it and go up... *[repeats gesture]*

In the student's explanation, he student sends a 20 Newton force "to stop" the ship. In the student's mind, the force tile does not represent a mere command for the ship to apply force and decelerate, but rather the force tile object itself travels into the simulation space to oppose the movement of the ship.

This distinction is important with regard to the student's developing epistemology of force. Within the framework of the force tile merely representing a command of the ship to apply force, the action of the ship carrying out the force tile's command represents a change in velocity to decelerate the ship, Newton's second law of motion. The student's conception of the force tile being 'sent' into the simulation space to 'oppose' the ship, however, gives agency to the force tile to travel into the simulation space and push backward on the ship in order to stop it, an enactment Newton's third law of motion. This unintended consequence is directly related to the design of the force representation.

The student's teacher, perhaps unsurprisingly, also echoes his student's epistemological misconception. Following gameplay, the student's teacher was given an example level from the game and asked to identify each of Newton's laws in the level:

Teacher: Newton's second... of course, when I change from at rest to in motion I've applied a force. So [the ship] starts moving from left to right. When I stopped [the ship] here I had to put an unbalanced force on it to go up to down.

Teacher: Newton's third law... opposites. When I stopped the ship I had to apply an opposite force of the same force amount to make my ship stop.

In these two statements, the teacher's epistemology of force becomes evident: unbalanced forces (Newton's second law) start the motion of the ship and opposing forces (Newton's third law) stop the ship. Parsing the teacher's response, the verb 'to apply' takes center stage. In his second law formulation, the teacher "applied a force" and in his third law formulation, the teacher also "had to apply an opposite force" in order to achieve the outcome he desired in the simulation space. Within the semantic frame of application, force is no longer applied by the ship, but by the teacher. What and where this force is, however, remains elusive. It is conceivable, based on the formulation of Newton's third law to 'stop the ship', that the ability to apply force in the simulation environment is a property of the force tile, which pushes on the ship to cause it to stop. As the teacher seeks to answer the question ‘What is force?’, the representations of the learning game lead to the conclusion that force is a property of an acting object opposing another acting object, scaffolded by the representation of the force tile opposing the ship.

3 Redesign Suggestions for Scaffolding Learning

As a result of the effects of representations on scaffolding epistemological formation evidenced in the student interview, two considerations for future design of scaffolding in digital learning environments emerge.

3.1 Exploration of Core Concepts

Confusion emerges on the part of the student as to the nature of force. Integrating opportunities within the game to explore the question “what is force?” could potentially clarify for students what the force tiles represent, allow for the representation to better scaffold understanding of force and motion, and further reinforce canonical understanding of Newton’s laws. In the absence of such an exploration, students are free to ascribe their own properties to the objects, ‘sending’ them to do work that they are actually incapable of doing.

3.2 Explicit Modeling of Representations

Beyond exploration, however, teachers and students must have the nature of representations in gaming environments explicitly modeled to ensure properties of the object are correctly ascribed. In the SURGE example, a simple statement that the force tiles are not, in fact, independent objects that travel to the simulation space and push on the ship, but rather are simply commands given to the ship to fire its rockets, could potentially alleviate the confusion as to the tile’s agency in the simulation space.

References

1. Clark, D., Nelson, B., Sengupta, P., & D’Angelo, C. (2009). Rethinking science learning through digital games and simulations: Genres, examples, and evidence. In Learning science: Computer games, simulations, and education workshop sponsored by the National Academy of Sciences, Washington, DC.
2. Ball, D. L., & Cohen, D. K. (1996). Reform by the book: What is: Or might be: The role of curriculum materials in teacher learning and instructional reform? *Educational Researcher*, *25*(9), 6-14.
3. Lehrer, R., & Schauble, L. (2002). Symbolic communication in mathematics and science: Co-constituting inscription and thought. In E. Amsel & J. Byrnes (Eds.), The development of symbolic communication (pp. 167-192). Mahwah, NJ: Erlbaum.

Fostering Diagnostic Accuracy in a Medical Intelligent Tutoring System

Reza Feyzi-Behnagh, Roger Azevedo, Elizabeth Legowski, Kayse Reitmeyer, Eugene Tseytlin, and Rebecca Crowley

Department of Educational and Counselling Psychology, McGill University, Montreal, Canada
Department of Biomedical Informatics and Pathology, University of Pittsburgh, PA, USA
reza.feyzibehnagh@mail.mcgill.ca; roger.azevedo@mcgill.ca;
{legoex, reitmeyerkl, tseytline, crowleyrs}@upmc.edu

Abstract. Diagnostic classification is an important part of clinical care, which is often the main determinant of treatment and prognosis. Clinicians' under- or over-confidence in their performance on diagnostic tasks can result in diagnostic errors which can lead to delay in appropriate treatment and unnecessary increase in the cost of medical care. This paper presents a version of SlideTutor aiming to reduce pathologists' and dermatopathologists' bias in diagnostic decision-making. This is accomplished by frequently prompting them to make metacognitive judgments of confidence, presenting them with the expert diagnostic solution path for each case, and de-biasing them by making them conscious of their metacognitive biases. This paper describes and summarizes the functionalities of SlideTutor, its cognitive training, tutoring phase, expert feedback, metacognitive intervention, and the open learner model.

1 Introduction and Background

Intelligent tutoring systems (ITSs) are adaptive and personalized instructional systems designed to mimic the well-known advantages of human one-on-one tutoring over other types of instructional methods [e.g., 1]. ITSs are capable of accelerating and enhancing the training of novices by providing adaptive and individualized scaffolding and feedback based on a complex interaction between several modules representing the domain knowledge as well as learner knowledge acquisition and development of expertise. The adaptive scaffolding and feedback in ITSs are targeted at improving student learning and fostering skills, such as making accurate metacognitive judgments [see 2]. In contexts where the teacher has limited time to spend on presenting content, teaching problem solving skills, and providing tailored feedback to individual students, ITSs can prove extremely helpful by providing adaptive individualized instruction to learners, organize content, and point out their errors for as much time and as many iterations as the learner requires [3].

ITSs can prove beneficial in training of highly specialized clinicians, such as pathologists. Training of specialized clinicians is very difficult in traditional training contexts for several reasons, including insufficient exposure to infrequently encountered cases, and the increased workloads of mentors which limit the time for training the next generation of practitioners and increase the potential for clinical errors among less-experienced practitioners. Training of pathologists typically requires five or more years, which includes both residency training (3-5 years) and advanced fellowship (1-3 years). In the context of training pathologists, ITSs could help alleviate many of the above-mentioned problems by providing a safe environment where residents can practice whenever they have time and as frequently as needed, and receive individualized feedback and guidance without inadvertently harming patients in the process. More specifically, ITSs can scaffold residents' accuracy of diagnoses, thereby alleviating their overconfidence or under-confidence in their performance on diagnostic tasks. Overconfidence would cause the clinician to conclude the diagnosis too quickly, therefore neglecting to fully consider alternative hypotheses and all the evidence in the case, which can result in diagnostic errors [4]. On the other hand, under-confidence might lead them to order unnecessary or inappropriate additional testing and use consultative services, which increases the risk of iatrogonic complications (i.e., complications caused by medical treatment or diagnostic procedures), delays treatment, and unnecessarily increases the costs of medical care [5].

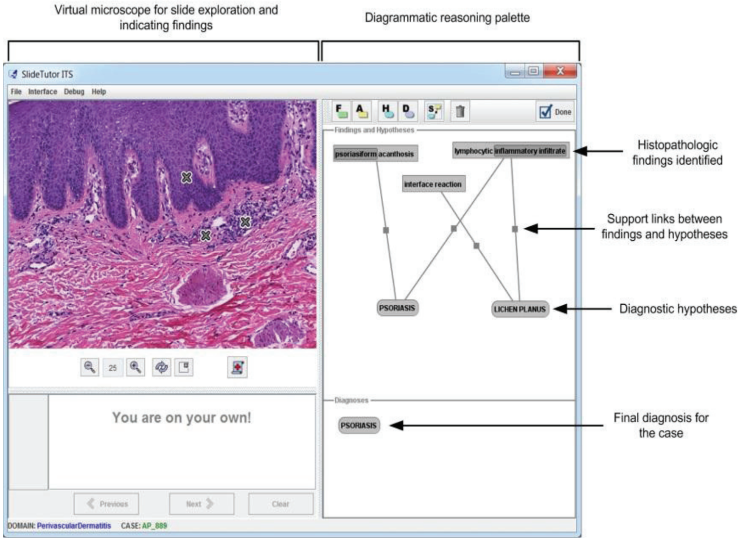
In order to alleviate the problem of under- or overconfidence in residents' diagnostic performance (i.e., poor calibration of judgment and performance), scaffolding needs to be provided to improve the accuracy of their metacognitive judgments (i.e., Feeling of Knowing, FOK) and eliminate any diagnostic bias. FOK is defined as the learner's certainty of his/her actual performance [6]. ITSs can play a significant role in assisting pathologists in making more accurate metacognitive judgments about their diagnostic decision-making and performance, and as a result make more accurate diagnoses.

One of the important methods of scaffolding and improving learners' metacognitive skills and performance is the use of open learner models (OLMs) in ITSs. A student model is an important part of an ITS which observes learner behavior and builds an individualized qualitative representation of her/his cognitive and metacognitive skills and gets updated in real-time during learners' interaction with the ITS [7]. Learner models are usually embedded in the ITS architecture and are not visible to the students, however, several researchers [e.g., 8] have investigated the benefits of allowing learners to access their learner model (OLM). Research has indicated that the mere displaying of visualizations of OLMs in ITS interfaces raises the awareness of the learners, allowing them to reflect on different aspects of their learning and problem solving. Besides all the advantages of using OLMs in interactive ITSs, according to [9], no study has investigated the use of OLMs for displaying metacognitive processes (e.g., metacognitive judgments of correctness of performance). In spite of the great potential and possibilities offered by the use of medical ITSs, few of these systems have been fully developed [e.g., 9] and only a fewer have been empirically evaluated [e.g., 10].

In this paper, we describe an adapted version of SlideTutor, an ITS which scaffolds pathology residents' accuracy of metacognitive judgments using different metacognitive interventions and an OLM for presenting metacognitive accuracy. The paper does not include our evaluation of the effectiveness of the implemented modules.

2 Description of the Medical ITS: SlideTutor

The SlideTutor intelligent tutoring system (http://slidetutor.upmc.edu) was modified for use in this study. The computational methods and the architecture of the original system have been previously published [11]. For the current study, the system uses a modular architecture implemented in the Java programming. SlideTutor provides users with cases to be solved under supervision by the system. Cases incorporate virtual slides, which are gigabyte size image files created from traditional glass slides by concatenating multiple images from a high resolution robotic microscope. Virtual slides are annotated using a custom built editing environment to produce case representations of discrete findings and their locations. A separate Ontology Web Language (OWL) based expert knowledge base consists of a comprehensive set of evidence-diagnosis relationship for the entire domain of study. A reasoning module uses a decision tree approach to construct a dynamic solution graph (DSG), representing the current state of the problem and all acceptable next steps including the best-next-step. As for the interface, participants use a graphical user interface (Fig. 1) to examine and diagnose the cases. Participants can pan and zoom in the virtual slide, locate findings using the mouse, and select from lists of findings and qualifiers, such as size and type, from a tree-like representation. Once findings are specified, they appear as evidence nodes in the diagrammatic reasoning palette (Fig 1). Afterwards, participants assert hypotheses using a separate tree-based menu, which eventually appear as nodes in the diagrammatic reasoning palette. Support links can then be drawn between evidence and hypothesis nodes to specify relationships between the two. Finally, one or more hypotheses may be dragged to the diagnosis window, and selected as the final diagnosis(es) before proceeding to the next case.


Virtual microscope for slide exploration and indicating findings
Diagrammatic reasoning palette
SlideTutor ITS
File Interface Debug Help
Done
Findings and Hypotheses
psoriasiform acanthosis
lymphocytic inflammatory infiltrate
interface reaction
PSORIASIS
LICHEN PLANUS
Diagnoses
PSORIASIS
25
Previous
Next
Clear
Histopathologic findings identified
Support links between findings and hypotheses
Diagnostic hypotheses
Final diagnosis for the case

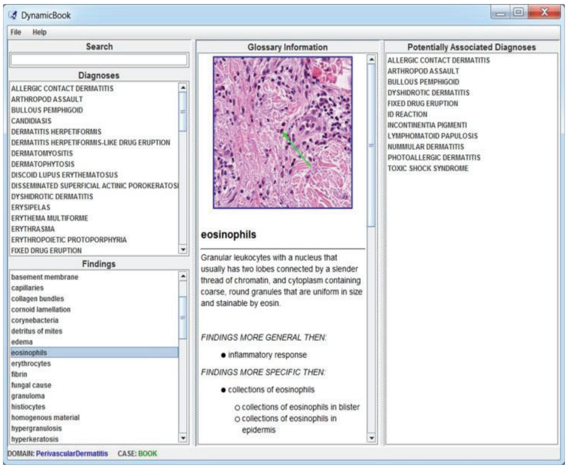
You are on your own!

DOMAIN: PerivascularDermatitis CASE: AP_889

Fig. 1. SlideTutor interface

2.1 The Dynamic Book

An interactive knowledge browser has been developed (called the Dynamic Book) that shows feature-diagnosis relationships as well as glossary information on all features and diagnoses in the selected domain of dermatopathology (i.e., perivascular diseases) (Fig. 2). A description of the domain and the cases is presented in the next section. A total of sixty-two diagnoses and fifty-seven findings are presented in this interface. Six of the diagnoses comprising six patterns were used in the tutoring phase of the study. By clicking on each one of the diagnoses, an image is presented in the interface showing an example of how the disease presents on a patient's skin. A description of the diagnosis was also presented under the image. Additionally, a list of potentially associated findings is presented to the right of the image and diagnosis description. A zoomed-in virtual slide image accompanied each of the findings in the list, where the presentation of the finding is indicated by an arrow. A description of the particular finding together with a list of potentially associated diagnoses is also presented. In order to guide the exploration of participants during the Dynamic Book phase towards important parts of the book, they are provided with a list of tasks to work through which pertained to a mix of patterns they would encounter in the tutoring phase and ones they would not.



ALLERGIC CONTACT DERMATITIS
ARTHROPOD ASSAULT
BULLOUS PEMPHIGOID
CANDIDIASIS
DERMATITIS HERPETIFORMIS
DERMATITIS HERPETIFORMIS-LIKE DRUG ERUPTION
DERMATOMYOSITIS
DERMATOPHYTOSIS
DISCOID LUPUS ERYTHEMATOSUS
DISSEMINATED SUPERFICIAL ACTINIC POROKERATOSI
DYSHIDROTIC DERMATITIS
ERYSIPELAS
ERYTHEMA MULTIFORME
ERYTHRASMA
ERYTHROPOIETIC PROTOPORPHYRIA
FIXED DRUG ERUPTION

basement membrane
capillaries
collagen bundles
cornoid lamellation
corynebacteria
detritus of mites
edema
eosinophils
erythrocytes
fibrin
fungal cause
granuloma
histiocytes
homogenous material
hypergranulosis
hyperkeratosis

eosinophils

Granular leukocytes with a nucleus that usually has two lobes connected by a slender thread of chromatin, and cytoplasm containing coarse, round granules that are uniform in size and stainable by eosin.

FINDINGS MORE GENERAL THEN:

- inflammatory response

FINDINGS MORE SPECIFIC THEN:

- collections of eosinophils
  - collections of eosinophils in blister
  - collections of eosinophils in epidermis

ALLERGIC CONTACT DERMATITIS
ARTHROPOD ASSAULT
BULLOUS PEMPHIGOID
DYSHIDROTIC DERMATITIS
FIXED DRUG ERUPTION
ID REACTION
INCONTINENTIA PIGMENTI
LYMPHOMATOID PAPULOSIS
NUMMULAR DERMATITIS
PHOTOALLERGIC DERMATITIS
TOXIC SHOCK SYNDROME

DOMAIN: PerivascularDermatitis CASE: BOOK

Fig. 2. Dynamic book interface

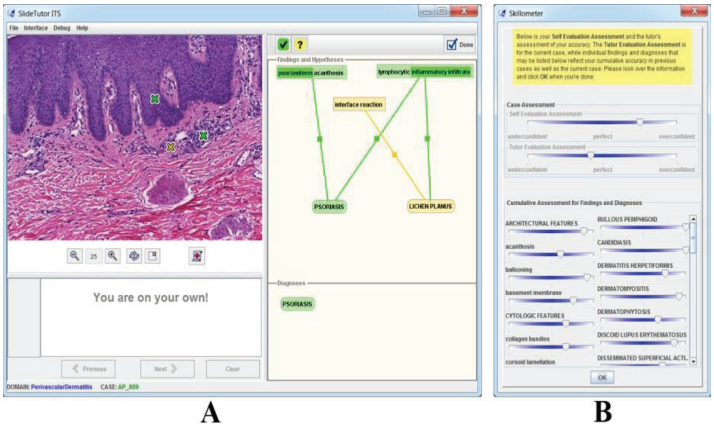
2.2. Pathology Cases

The Perivascular Dermatitis domain was selected for the current SlideTutor study because the domain is well-tested, includes patterns (i.e., a combination of evidence identified in a particular case) with multiple cases, and more cases are available than other domains. Also, Perivascular Dermatitis is a large domain and it is unlikely that participants would have complete knowledge of this diagnostic area. 20 cases were used for the tutoring phase. Cases were obtained from the University of Pittsburgh

Medical Center (UPMC) slide archive and from private slide collections. Diagnoses were checked and confirmed by a dermatopathologist prior to inclusion in the system repository. For each case, a knowledge engineer and an expert dermatopathologist collaborated in defining all present and absent findings, their locations on the slide (case annotation), and relationships among findings and diagnoses (knowledge-base development). Each diagnosis included a set of one or more diseases that matched the histopathologic pattern.

2.3 The Coloring Book and Metacognitive Judgments

For the intervention condition, once participants complete identifying findings, hypotheses, and diagnoses for a case, they progress to an interface called the Coloring Book (Fig. 3A). In this interface, they indicate if they are sure or unsure of the items they identified for the case (i.e., FOK judgments) by clicking on them and coloring them as either green (sure) or yellow (unsure). Next, they are presented with a window with a slider where they indicate how accurate they think their self-assessments in the coloring book were (ranging from underconfident to overconfident). Afterwards, they are presented with correct findings, hypotheses, and diagnoses for the respective case (colored in green) and incorrectly identified items as red. After reflecting on their performance and the feedback from the system, they are presented with a window juxtaposing the sliders for their self-assessment of their FOK judgments and the evaluation of the tutor based on their performance and their FOK judgments (the open learner model: OLM) (Fig 3B). At the bottom of the window, one or more individual findings or diagnoses may be listed, which reflects the participant's cumulative accuracy in previous cases as well as the current case for the particular finding or diagnosis. At the end, they are asked to make another metacognitive judgment and state whether they would feel confident solving similar cases, to which they respond on a 6-point Likert scale ranging from "not confident" to "very confident". This concludes the case, and progresses them to the next case.



You are on your own!

Below is your **Self Evaluation Assessment** and the tutor's assessment of your accuracy. The **Tutor Evaluation Assessment** is for the current case, while individual findings and diagnoses that may be listed below reflect your cumulative accuracy in previous cases as well as the current case. Please look over the information and click **OK** when you're done.

Fig. 3. Coloring book interface (A) and the OLM (B)

3 Study Timeline

As part of the design of the study and interface of the ITS, the study phases and timeline were determined as follows (Fig. 4). An approximate total time of four hours was allocated as the participant session time. At the beginning and after signing the informed consent form, the participants were administered a test (pre-pre-test) of their prior knowledge of the domain targeted by the current version of SlideTutor (i.e., Perivascular diseases). Next, they spent 30 minutes acquiring cognitive knowledge of the domain while accomplishing a task given to them by the experimenter (Dynamic Book phase). Afterwards, another test of cognitive knowledge of the domain was administered (pre-test). Once the test was completed, they proceeded to the tutor training and tutor use phase (in intervention or control condition) where they solved 20 cases and indicated their confidence in their responses and were shown an OLM (intervention condition), or solved the cases and progressed with no feedback from the system (control condition). At the end, a post-test was administered to gauge their knowledge gains during interactions with the tutor. A detailed description of the ITS, the tests, dynamic book, and the tutoring interventions is presented below.

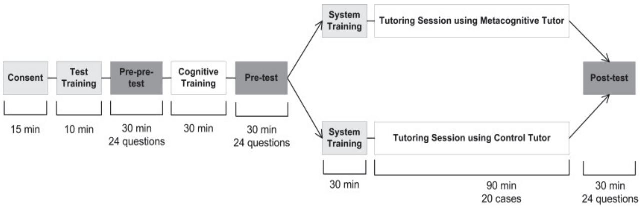


Fig. 4. Study timeline

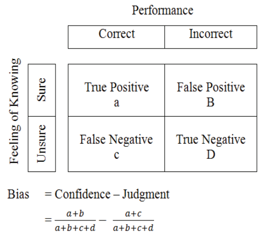
4 Measures

4.1 Cognitive Measures

In order to measure the prior cognitive knowledge of the domain at the beginning of the tutoring session, cognitive gains after the cognitive learning phase, and the knowledge gains after the tutoring session, three 24-item tests were administered. Three versions of each test were created, and the test order was randomized per session to control for order effects. Each test comprised of 24 questions, and the questions were a mix of tutored and untutored items. Tutored items were about the material that was presented in the cases seen with the tutoring system, while untutored items were about material that was not covered by the tutoring system. Three question types were used in the tests: finding, diagnosis, and differentiate questions. Finding questions consisted of a static microscopic image with an arrow pointing at a feature to be identified. Diagnosis questions consisted of a list of findings, and participants had to provide the diagnosis(es) that match the findings. Differentiate questions consisted of two diagnoses, and participants had to provide a feature that can be used to differentiate the two. After responding to each question, participants were asked to rate if they were sure or unsure of their responses using radio buttons (FOK metacognitive judgment).

4.2 Metacognitive Measures

Feeling of knowing (FOK) metacognitive judgment measures were collected on all test items in the three cognitive knowledge tests and on all findings, hypotheses, and diagnoses identified in cases in the tutoring phase. The FOK measures were collected as binary values: sure vs. unsure. The data from metacognitive ratings on test questions were only used for analyses after the study was completed. However, the metacognitive judgment ratings for items identified in cases in the tutoring phase in the Coloring Book layout (see section 2.3) were used for calculation of a measure of over- or under-confidence called Bias, which was presented to the participant after solving the case and indicated their confidence in the items they identified in the case (in the OLM: see section 2.3). The bias score is calculated by subtracting the relative performance on all items (total correct items divided by all items) from the proportion of items judged as known (total sure items divided by all items) [12]. Figure 5 indicates how bias scores are calculated. Positive bias scores indicate over-confidence and negative scores indicate under-confidence. When performance perfectly matches the rated confidence level, the bias score equals zero. In other words, the bias score indicates the direction and degree of lack of fit between confidence and performance [13]. The bias score for each case was presented to the participant in the form of a slider ranging from under-confident to perfect to over-confident with a cursor indicating the participant's bias score.



Performance

Feeling of Knowing		Correct	Incorrect
	Sure	True Positive a	False Positive B
	Unsure	False Negative c	True Negative D

Bias = Confidence – Judgment

$$= \frac{a+b}{a+b+c+d} - \frac{a+c}{a+b+c+d}$$

Fig. 5. FOK contingency table and the calculation of bias

5 Conclusion

We described the functionalities of a version of SlideTutor aimed at reducing the metacognitive bias of pathologists and dermatologists while diagnostic decision-making by deploying metacognitive interventions and using an open learner model to aid participants in reflecting on their diagnostic performance. Open learner models have not been used in the previous studies for displaying the metacognitive performance of participants [8], and the current iteration of SlideTutor is novel in this regard. The Dynamic Book interface used for the cognitive learning phase provided participants with an environment to conduct a targeted search and knowledge acquisition (targeted at completing the task assigned by the experimenter). As mentioned above, since the domain chosen for this version of SlideTutor is a very large domain, a cognitive learning phase was deemed necessary in order to provide the opportunity for acquisition of some cognitive knowledge and freely explore the glossary of diagnoses and findings.

Acknowledgment

The authors gratefully acknowledge the support of this research by the National Library of Medicine through grant number 5R01LM007891.

References

1. Koedinger, K., Aleven, V., Roll, I., & Baker, R. (2009). In vivo experiments on whether supporting metacognition in intelligent tutoring systems yields robust learning. In A. Graesser, J. Dunlosky, D. Hacker (Eds.), *Handbook of metacognition in education*. 383-413. Mahwah, NJ: Erlbaum.
2. Azevedo, R., & Hadwin, A. F. (2005). Scaffolding self-regulated learning and metacognition: Implications for the design of computer-based scaffolds. Instructional Science, 33, 367-379.
3. VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. Educational Psychologist, 46(4), 197-221.
4. Graber, M. L., Franklin, N., & Gordon, R. (2005). Diagnostic error in internal medicine. Archives of Internal Medicine, 165 (13), 1493-1499.
5. Berner, E. S. & Graber, M. L. (2008). Overconfidence as a cause of diagnostic error in medicine. The American Journal of Medicine, 121 (5A), S2-S23.
6. Metcalf, J., & Dunlosky, J. (2008). Metamemory. In H. Roediger (Ed.), Cognitive psychology of memory (Vol. 2, pp. 349-362). Oxford: Elsevier.
7. Bull, S. (2004). Supporting Learning with Open Learner Models. Proceedings of 4th Hellenic Conference with International Participation: Information and Communication Technologies in Education, Athens, Greece. Keynote.
8. Bull, S., & Kay, J. (in press). Open learner models as drivers for metacognitive processes. In R. Azevedo & V. Aleven (Eds.). International handbook of metacognition and learning technologies. Amsterdam, The Netherlands: Springer.
9. Azevedo, R., & Lajoie, S. (1998). The cognitive basis for the design of a mammography interpretation tutor. International Journal of Artificial Intelligence in Education, 9, 32-44.
10. El Saadawi, G. M., Tseytlin, E., Legowski, E., Jukic, D., Castine, M., Fine, J., . . . Crowley, R. S. (2008). A natural language intelligent tutoring system for training pathologists: implementation and evaluation. Advances in Health Sciences Education: Theory and Practice, 13, 709-722.
11. Crowley, R. S., & Medvedeva, O. (2006). An intelligent tutoring system for visual classification problem solving. Artificial Intelligence in Medicine, 36 (1), 85-117.
12. Kelemen, W. L., Frost, P. J., & Weaver III, C. A. (2000). Individual differences in metacognition: Evidence against a general metacognitive ability. Memory & Cognition, 28(1), 92-107.
13. Schraw, G. (2009). A conceptual analysis of five measures of metacognitive monitoring. Metacognition and Learning, 4(1), 33-45.

Teacher Perspectives on the Potential for Scaffolding with an Open Learner Model and a Robotic Tutor

Aidan Jones, Susan Bull and Ginevra Castellano

Electronic, Electrical and Computer Engineering, University of Birmingham, UK
axj100@bham.ac.uk, s.bull@bham.ac.uk, g.castellano@bham.ac.uk

Abstract. This paper considers the potential for scaffolding learning in open-ended learning environments using a robotic tutor and an open learner model. While we expect this approach to be more broadly applicable, we here illustrate with a map-reading activity in geography and/or environmental sciences. The paper presents issues raised in open-ended teacher interviews, which suggest real possibilities for incorporating a robotic tutor together with an open learner model in the classroom.

Keywords: affect detection, open learner model, scaffolding, social robotics

1 Introduction

Open learner models (OLM) externalise the learner model in a way that is interpretable by the user, e.g.: skill meters [16], concept maps [19], treemaps [14]. One of the aims of opening the learner model to the learner is to help promote reflection on the part of the learner; to facilitate their planning and decision-making; and raise their awareness of their understanding or their developing skills [3]. Thus, the OLM can be seen as a form of scaffolding for cognitive and metacognitive processes, with a particular focus on supporting and developing self-regulation. This focus is very much in line with previous considerations of tools offering scaffolding (see e.g. [1]). This approach to supporting the learner can be very light or can be more closely guided, depending on the level of detail of the modelling and the visualisation of the model, as well as the goals of the interaction and the user's current learning needs.

Most learner models that are inspectable by the learner have focussed on knowledge-related attributes. However, despite it being a difficult task, there is growing interest in detecting and responding to affective states (e.g. [6]; [24]; [25]), and increasingly with a goal of adaptive scaffolding to support individual differences [10]. A taxonomy of "academic emotions, which are directly related to academic learning, classroom instruction or achievement", has been identified [17]: the positive activating emotions of enjoyment, hope, and pride; the positive deactivating emotion of relief; the negative activating emotions of anger, anxiety, shame; and the negative deactivating emotions of hopelessness and boredom.

OLMs can offer an additional mechanism by which learner model data about affective states can be confirmed and/or clarified. In addition to visualisation of the learner model, the term ‘open learner modelling’ encompasses methods that allow users to contribute to, edit, or negotiate the contents of the learner model [3]. While we do not wish to require or rely on self-report about emotions and affective states, if a learner is frustrated by feedback that has been generated in part based on inaccurate or incomplete affect detection, a simple method to advise a learning environment of this could be of substantial benefit. Thus, while providing an OLM of the more traditional knowledge/skills representations, we recommend also allowing the learner to access the representations regarding their affective state (e.g. inferred through sensors [24], semantic and contextual cues [25], or based on a video corpus of affective expressions [7]). This may bring new issues to the problems of affect modelling (e.g. if the learner model indicates an affective state that the learner disagrees with, might this make them angry, demotivated or frustrated?) Nevertheless, as well as offering an opportunity to modify or influence the representation of affect, it may also help increase learner trust in the learner model, as the user will be able to identify why certain aspects of feedback or scaffolding are tailored in the manner that they are, and have the opportunity to address or challenge any discrepancies. In this paper we take the starting point of benefits previously demonstrated for OLMs (e.g. [12]; [16]), and consider their use in a more open-ended context, and with affect modelling.

2 Scaffolding with an Open Learner Model

As argued above, OLMs can be considered as ways to help scaffold learning and the learning process, and may have particular potential in open-ended tasks and environments. With the increasing focus on professional competency frameworks and the inevitable extension of the competency perspective to educational contexts (e.g. for language [8], for STEM literacy [2], for geography [21]), there comes even greater scope for future use of open-ended learning environments, and corresponding challenges for scaffolding learning in such situations. Competency frameworks have already been applied in a generic OLM context, with examples for language [4] and meeting facilitation [20]. We propose that such approaches be further developed to meet the requirements of the changing educational focus, curricula, and assessment.

We illustrate here with a geography and/or environmental science map-based activity, where tools may be used to discover information from a map, to measure distance and area, to view terrain or entities on the map such as buildings, cities and countries. The learner may identify features, follow directions in a trail, explore the area, or determine the best location for some purpose (e.g. where to situate a new visitor centre). Such activities can range from specific to very open-ended, and a range of competencies may be demonstrable (e.g. map-reading, map sketching, mapping, geographical argumentation, ethical judgement (see [21]).) This relates closely to the England and Wales National Curriculum for Geography [9] key processes, e.g.:

“Pupils should be able to:

- use atlases, globes, maps at a range of scales, photographs, satellite images and other geographical data;
- ask geographical questions, thinking critically, constructively and creatively;

- analyse and evaluate evidence, presenting findings to draw and justify conclusions;
- solve problems and make decisions to develop analytical skills and creative thinking about geographical issues."

However, the nature of this type of open-ended activity may also lead to different affective states across and within individuals. In the next section we consider the opportunities for improving scaffolding using OLMs that include representations of affective states, supported by an empathic robotic tutor.

3 Support from a Robotic Tutor

Opening up a system's representations of a learner's affective state could, as indicated above, further influence learner affect. To mitigate a possibly negative reaction that could impact motivation, we recommend taking a social robotics approach. Artificial tutors may incorporate their understanding of the learner's emotional state in their pedagogical strategies and interventions [5]. The presence of a 2D or 3D character has revealed some positive learning effects, especially in engagement [15]; and recall has been shown to be higher with a robotic teacher when adaptive cues have been given based on EEG measurements of engagement [22]. Studies that compared virtual representations of characters with robots showed a preference for robotic embodiment with reference to social presence [13], enjoyment [18] and performance [11]. Thus, we suggest this to be a useful avenue to explore for scaffolding learning particularly when affective states are also modelled. For example, Figure 1 shows the Nao Robot and its ability to point or gesture towards items on a tabletop, which include visualisations of the learner model. Since many of the activities we envisage are map-based, we will use an interactive map approach on a touch table in this example.

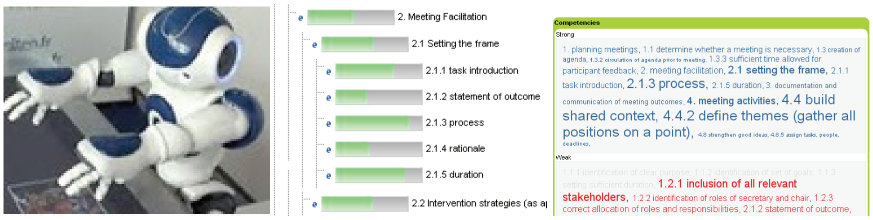

2. Meeting Facilitation
2.1 Setting the frame
2.1.1 task introduction
2.1.2 statement of outcome
2.1.3 process
2.1.4 rationale
2.1.5 duration
2.2 Intervention strategies (as a|
Competencies
Strong
1. planning meetings, 1.1 determine whether a meeting is necessary, 1.3 creation of agenda, 1.3.2 circulation of agenda prior to meeting, 1.3.3 sufficient time allowed for participant feedback, 2. meeting facilitation, 2.1 setting the frame, 2.1.1 task introduction, 2.1.3 process, 2.1.5 duration, 3. documentation and communication of meeting outcomes, 4. meeting activities, 4.4 build shared context, 4.4.2 define themes (gather all positions on a point), 4.8 strengthen good ideas, 4.8.5 assign tasks, people, deadlines,
Weak
1.1.1 identification of clear purpose, 1.1.2 identification of set of goals, 1.1.3 setting sufficient duration, 1.2.1 inclusion of all relevant stakeholders, 1.2.2 identification of roles of secretary and chair, 1.2.3 correct allocation of roles and responsibilities, 2.1.2 statement of outcome,

Fig. 1. The Nao Robot and a competency-based open learner model (skill meters and word cloud shown, from the Next-TELL open learner model [20])

Examples of general interactions and scaffolding between the learner and the robot include: offering assistance by guiding the learner through instructions; asking questions (to prompt reflection); gestures (to illustrate or focus attention, or indicate shared focus); offering affective support if learners' actions are not optimal (telling them not to worry and try again); drawing attention back to task if a learner becomes distracted; mirroring affective state when this is positive, and bringing awareness to affective state if it is negative. This aims to foster a perception of the robot as empathic (see e.g. [7]).

In addition to the learner model visualisations on the tabletop, the robot can itself express the model content by giving a summary of relevant knowledge or competencies, perhaps at the start of a session to show that it remembers the learner, but also during a session to give the learner a sense of achievement and to prompt them to think about their learning and how they might use the learner model information. As with adaptive scaffolding in general, interaction about the learner model will be tailored as appropriate to the individual, as will other scaffolding behaviours from the robot.

When using the OLM to investigate its representations of their affective state, the learner will already be accustomed to the robot's shared understanding of their competencies. Therefore, when it then comes to reviewing affective states in the model, the robot's ability to invite or allow discussion or adjustment to the affective model contents can build on the relationship that the learner has with the robot, with reference to their understanding or competencies. This approach will build on previous findings using a chatbot, that child-system negotiation of the knowledge-focussed data in an OLM resulted in significant improvements in children's learning without additional tutoring [12]. In that case negotiation involved student or system challenges and discussion about the child's beliefs (representations in the learner model) with the aim of prompting reflection and increasing the accuracy of the learner model by taking students' opinions about their learning into consideration. In our current work we propose also encouraging the learner to think about their affective state, how this may influence their learning, and how they might regulate their affect. In effect, this is an approach to help learners self-scaffold during the transition from more tightly to less tightly guided interaction. The first step towards this goal involves obtaining teacher viewpoints on the potential of this approach in the classroom. This is considered in the next section.

4 Teacher Interviews

Following from the arguments above that suggest possibilities for scaffolding in open-ended tasks using an OLM together with an empathic robot, teacher interviews were undertaken to determine the likelihood of uptake of this approach in contexts where the required technologies are already in place.

4.1 Participants, Materials and Methods

Seven participants took part in open interviews (4 teachers, 2 teaching assistants, 1 trainee). The aims of the study were described, highlighting emphatic tutoring and interaction, and personalising robotic tutoring to the learner's needs. In a semi-formal interview, specific questions relevant to scaffolding and OLMs included:

- What role would a system like this play? (To ascertain teachers' views on how the robot could effectively 'fit' into the classroom and classroom practice.)
- If you had a robot that could monitor how a child is progressing, how would you like that robot to interact with the child? (To provide information for the design of the learning scenarios and robot interactions.)
- Would it be beneficial to set the level of difficulty and how do you do this at the moment? (To gauge the extent of teachers' likely acceptance of a coarse-grained personalisation approach with a robotic tutor.)

- How do you detect when a student is having difficulties and how do you help the learner overcome the difficulties? (To ascertain how teachers detect when a learner is facing difficulties in this kind of open-ended activity, and whether they may be receptive to more fine-grained adaptation with the robotic tutor.)

Written notes were made by the researcher. Comments were then categorised to help design subsequent formal interviews before building the prototype environment.

4.2 Results

Table 1 summarises the number of teachers expressing each of the points addressed below, following the comment categorisations, with representative viewpoints then discussed further. Several teachers were very interested in the fact that they could use the system to encourage independent learning, as this is becoming a key objective for teachers. To address the varied needs of students, at the moment the teachers might give out different question sheets to different students. Typically the teachers change the difficulty of an activity by changing the language style, the number of prompts, breaking down the activity into smaller steps, and the amount of scaffolding provided. The most difficult questions or problems may be very open-ended, and require the learner to argue a point in their own words, or the teacher may apply extra constraints such as working within a budget. All teachers were keen that the system to be trialled should be able to respond to the individual, stretching the most able while also ensuring suitable personalisation for the less advanced students.

Table 1. Teacher comments categorised

Comment	**No. teachers**
Encourage independent learning	3
Personalisation / adaptivity	7
More open-ended activities	7
Prompt metacognitive behaviours	7
Affect detection	7
Use of progress bars	2
Incorporation of robot into classroom	7

In addition, all teachers stated that they would like the learning activities they undertake to easily move beyond basic map reading skills to activities where the learner needs to make comparisons, decisions and arguments. Comparisons in this space could be to compare high and low CO2 production, population density, and similar. Decisions and arguments could be made on tasks which involve, for example, deciding on the most appropriate location for a visitor centre or flood defence: the learner must make an argument in favour or against an action. Thus, the teachers are looking for ways to incorporate more open-ended activities into the classroom interaction. All also wanted to encourage reflection and metacognitive behaviours, for example, by saying “Have a think”, “Did you consider...?”. They also thought that the robot could usefully point out that there is no really wrong answer in some of the activities.

All teachers already detect whether a learner is having difficulties, from their behaviour. For example: the teacher can tell if the learner is not listening, not paying attention or not understanding. This information can come from their facial expression, where they are looking, whether they are fidgeting, how they respond to instructions, whether they are actively asking for help verbally or by raising their hand, or if they are chatting or disrupting other children. The teachers can also identify whether a learner is attempting a task in a sub-optimal way.

Two teachers suggested that progress bars may be beneficial. They stressed that real time assessment would be desirable, and if a learner faced difficulties, these need to be caught promptly and acted upon as appropriate by the system or the teacher.

There were no concerns from any of the teachers about fitting the robot into the classroom activities, particularly if the lesson plan actively included the robot (e.g. as a station in a station rotation lesson where a number of learners in a class would have a turn with the robot). The teachers were interested in monitoring the learner's progress from a console, enabling the teacher to intervene if the learner stopped making progress, particularly useful if there were multiple learners interacting with multiple artificial tutors. They also thought that the simple fact that there was a robot would make any task seem novel and more engaging.

4.3 Discussion

Because the interviews were open, not all points were discussed in each interview. The lower level of comments in some areas therefore may not indicate disagreement, but rather that these issues were not raised during the interview.

The possibility for the robot to adapt to individuals, as requested by all participants, is exactly the kind of approach enabled by a learner model. For this reason the learner model is anticipated to be acceptable to teachers in this robot-tabletop context. All teachers also wished to use open-ended tasks such as described above, to match the requirements of the England and Wales National Curriculum for Geography [9]. This is, therefore, another indication of likely acceptance. Furthermore, because teachers are already identifying student engagement and other affective states, the modelling of affect and use of a physical robot is an approach that they will understand: while they may not be able to discuss knowledge and competencies individually to the extent they wish, a robotic tutor can help in this task while maintaining an approximation of the empathic approach a teacher would use. The fact that two teachers suggested progress bars indicates that these participants wish to have a view of learning visible on the tabletop, in line with OLM. In addition, the OLM should facilitate the kind of metacognitive behaviours considered important by all teachers. The request for being able to monitor learners is also in line with OLM also being a tool to support teachers [20]. This goes beyond many learning analytics visualisations (e.g. dashboards [23]), to focus on understanding, competencies, and now also affect.

An important immediate concern is practical deployment in the existing learning context and curriculum. All teachers could see how the robot and touch table could be integrated into the classroom, and could identify benefits for doing so. Thus we argue that there is a role for empathic robots and OLMs in scaffolding open-ended learning.

5 Summary and Conclusions

This paper has argued the benefits of using an OLM as a means to lightly scaffold learners in open-ended learning contexts where the development of self-regulation skills and metacognitive behaviours are considered important. This is becoming increasingly central with the competency focus adopted in many subjects and countries. Affective modelling is considered beneficial in such contexts, given the potential frustrations of the open-ended nature of activities, and the provision of a means to discuss and possibly correct the system representations of affect is suggested. Because of the advantages of robotic tutors, an empathic robot approach is proposed. The teacher interviews confirmed the feasibility of introducing this solution to real classrooms that have the appropriate technologies.

Acknowledgements

This work was partially supported by the European Commission (EC) and was funded by the EU FP7 ICT-317923 project EMOTE. The authors are solely responsible for the content of this publication. It does not represent the opinion of the EC, and the EC is not responsible for any use that might be made of data appearing therein. We thank the participating teachers from Haybridge High School and Stourport High School, Worcestershire, UK.

References

1. Azevedo, R. Using Hypermedia as a Metacognitive Tool for Enhancing Student Learning? The Role of Self-Regulated Learning, Educational Psychologist 40(4), 199-209 (2005)
2. Bybee, R.W.: Advancing STEM Education: A 2020 Vision, Technology and Engineering Teacher 70(1), 30-35 (2010)
3. Bull, S. & Kay, J.: Open Learner Models, in R. Nkambou, J. Bordeau & R. Miziguchi (eds), Advances in Intelligent Tutoring Systems, Springer, 318-338 (2010)
4. Bull, S., Wasson, B., Kickmeier-Rust, M., Johnson, M.D., Moe, E., Hansen, C., Meissl-Egghart, G. & Hammermuller, K.: Assessing English as a Second Language: From Classroom Data to a Competence-Based Open Learner Model, International Conference on Computers in Education (2012)
5. Burleson, W. Affective Learning Companions: Strategies for Empathetic Agents with Real-Time Multimodal Affective Sensing to Foster Meta-Cognitive and Meta-Affective Approaches to Learning, Motivation, and Perseverance, PhD Thesis, MIT (2006)
6. Calvo, R.A. & D'Mello, S.: Affect Detection: An Interdisciplinary Review of Models, Methods, and Their Applications, IEEE Transactions on Affective Computing 1(1), 18-37 (2010)
7. Castellano, G., Leite, I., Pereira, A., Martinho, C., Paiva, A. & McOwan, P/.W.: Multimodal Affect Modelling and Recognition for Empathic Robot Companions, International Journal of Humanoid Robotics 10(1) (2013)
8. Council of Europe.: The Common European Framework of Reference for Languages, http://www.coe.int/t/dg4/linguistic/Cadre1_en.asp. Accessed 18 March 2013.

9. Department for Education. Secondary National Curriculum until 2014 (Geography), http://www.education.gov.uk/schools/teachingandlearning/curriculum/secondary/b00199536/geography. Accessed 29 April 2013.
10. Desmarais, M.C. & Baker, R.J.D.: A review of Recent Advances in Learner and Skill Modeling in Intelligent Learning Environments, User Modeling and User-Adapted Interaction 22, 9-38 (2012)
11. Hoffmann, L. & Krämer, N.C. How Should an Artificial Entity be Embodied? Comparing the Effects of a Physically Present Robot and its Virtual Representation, Proceedings of Workshop on Social Robotic Telepresence, HRI (2011)
12. Kerly, A. & Bull, S.: Children's Interactions with Inspectable and Negotiated Learner Models, in B.P. Woolf, E. Aimeur, R. Nkambou & S. Lajoie (eds), Intelligent Tutoring Systems, Springer-Verlag, Berlin Heidelberg, 132-141 (2008)
13. Kidd, C. D.: Sociable Robots: The Role of Presence and Task in Human-Robot Interaction (Doctoral dissertation, Massachusetts Institute of Technology) (2003)
14. Kump, B., Seifert, C., Beham, G., Lindstaedt, S.N. & Ley, T. Seeing What the System Thinks You Know Visualizing Evidence in an Open Learner Model, LAK, ACM. (2012)
15. McQuiggan, S.W. & Lester, J.C. Modeling and Evaluating Empathy in Embodied Companion Agents, International Journal of Human-Computer Studie, 65, 348–360 (2007)
16. Mitrovic, A., Martin, B.: Evaluating the Effect of Open Student Models on SelfAssessment. International Journal of Artificial Intelligence in Education 17(2), 121--144 (2007)
17. Pekrun, R., Goetz, T., Titz, W. & Perry, R.: Academic Emotions in Students' Self- Regulated Learning and Achievement: A Program of Qualitative and Quantitative Research, Educational Psychologist 37(2), 91-105 (2002)
18. Pereira, A., Martinho, Leite, C.I. & Paiva, A. iCat the Chess Player: the influence of embodiment in the enjoyment of a game, Proceedings of 7th International Joint Conference on AAMAS , Estoril, Portugal, 1253-1256 (2008)
19. Perez-Marin, D., Alfonseca, E., Rodriguez, P., Pascual-Neito, I.: A Study on the Possibility of Automatically Estimating the Confidence Value of Students' Knowledge in Generated Conceptual Models, Journal of Computers 2(5), 17-26 (2007)
20. Reimann, P., Bull, S. & Ganesan, P.: Supporting the Development of 21st Century Skills: Student Facilitation of Meetings and Data for Teachers, in R. Vatrapu, W. Halb & S. Bull (eds), Proceedings of the Workshop Towards Theory and Practice of Teaching Analytics, EC-TEL 2012, CEUR Workshop Proceedings http://ceur-ws.org/Vol-894 (2012)
21. Rempfler, & Uphues, R.: System Competence in Geography Education Development of Competence Models, Diagnosing Pupils' Achievement, European Journal of Geography 3(1), 6-22 (2012)
22. Szafir, D. & Mutlu, B. Pay Attention! Designing Adaptive Agents that Monitor and Improve User Engagement, Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems ACM, 11-20 (2012)
23. Verbert, K., Duval, E., Klerkx, J., Govaerts, S., Santos, J.L.: Learning Analytics Dashboard Appliclations, American Behavioral Scientist, early online version Feb 28, DOI: 10.1177/0002764213479363, (2013)
24. Woolf, B.P., Arroyo, I., Cooper, D., Burleson, W. & Muldner, K.: Affective Tutors: Automatic Detection of and Response to Student Emotion, in R. Nkambou, J. Boureau & R. Mizoguchi (eds), Advances in Intelligent Tutoring Systems, Springer-Verlag, Berlin Heidelberg, 207-227 (2010)
25. Zhang, L.: Exploration of Affect Detection Using Semantic Cues in Virtual Improvisation, in S.A. Cerri, W.J. Clancey, G. Papadourakis & K. Panourgia (eds), Intelligent Tutoring Systems, Springer-Verlag, Berlin Heidelberg, 33-39 (2012)

Metacognitive Tutoring for Scientific Modeling

David A. Joyner, Ashok K. Goel, and David M. Majerich

Design & Intelligence Laboratory, School of Interactive Computing
Georgia Institute of Technology, Atlanta, Georgia, USA
david.joyner@gatech.edu; ashok.goel@cc.gatech.edu;
dmajerich6@mail.gatech.edu

Abstract. In this paper, we present a set of metacognitive tutors for teaching scientific inquiry-driven modeling. We describe the MeTA architecture in which the tutors are implemented and experiences with an initial pilot study.

Keywords: metacognitive tutors, intelligent tutoring systems, scientific modeling, middle school science.

1 Introduction

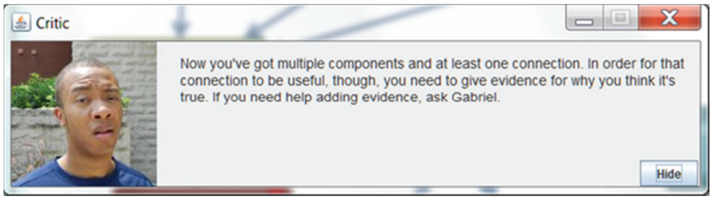
Supporting metacognition has been identified as one of the most important principles of instructional design [4]. In recent years, interventions using a variety of metacognitive skills have been studied. Aleven et al. examine the use of a metacognitive tutor for help seeking within a cognitive tutor for geometry [1]. Some systems, such as MetaTutor, focus on teaching students self-assessment skills to identify knowledge gaps or monitor their own progress [3,10]. Betty's Brain can teach students metacognitive skills by having them request that Betty engage in those skills herself [11]. These projects have shown the success of tutoring interventions based on developing metacognitive skills.

Inquiry-based learning has long been pursued as a desirable approach to classroom curriculum design [6], and significant efforts have been made to incorporate authentic scientific modeling and inquiry into science education, such as in projects like Thinker Tools [13]. This paper presents our early efforts to construct a metacognitive tutoring system specifically aimed at teaching these skills within an open-ended learning environment named MILA (for Modeling & Inquiry Learning Application).

2 Tutoring Scientific Inquiry-Driven Modeling in MILA

MILA (**M**odeling & **I**nquiry **L**earning **A**pplication) is an interactive learning environment for supporting learning about ecosystems in middle school science. Students use MILA to construct Component-Mechanism-Phenomenon models of complex ecological phenomena. Component-Mechanism-Phenomenon models are adaptations of Structure-Behavior-Function models [7,12], and MILA evolves from our earlier work on learning Structure-Behavior-Function models of ecosystems [8,12].

To support students' modeling and inquiry while engaging with MILA, we constructed a metacognitive tutoring system consisting of four separate metacognitive tutoring agents playing four different functional roles: a Guide, a Critic, a Mentor, and an Interviewer. Broadly, these tutors were constructed according to lessons and guidelines transferred from other initiatives in metacognitive tutoring [2,10]. Students interact with tutors by clicking tutors' avatars in the tutor box. Upon clicking, the tutor's window appears and gives the student any feedback it has available, as shown in Figure 2. Reactive tutors checks their Mappings when the student clicks in order to respond to students' help-seeking behaviors [1]. A proactive tutor actively monitor students' progress and interrupt the students to provide their feedback or ask their question in order to facilitate just-in-time error correction [10].



Now you've got multiple components and at least one connection. In order for that connection to be useful, though, you need to give evidence for why you think it's true. If you need help adding evidence, ask Gabriel.

Fig. 2. An example of one of the four tutors, the Critic. All tutors appear in dialog boxes such as this one. In addition to text feedback, tutors may ask students to answer questions or offer students questions they might want answered.

The Guide serves to answer students' questions, and thus is a reactive tutor. She is developed to anticipate what questions students may want to ask based on the current lesson, the students' current model, software, and tutor interactions and then offer those questions when called. For example, early in the unit, the Guide anticipates questions that largely focus on interaction with the software itself. Later, she expects and offers questions based on students' current models or recent model construction process.

The Critic analyzes students' models, validating students' models against a set of defined model criteria. He is a reactive tutor who only checks models when students are looking for feedback, demonstrating the knowledge gaps of which students should be aware in model construction and providing guidance on how to fill those knowledge gaps, as well as avoid them in the future.

The Mentor leverages the notion of cognitive apprenticeship [5]. He is a proactive tutor who observes students' interaction with the software and demonstrates new or difficult concepts. In practice, the main role of the Mentor has been to set expectations and learning goals, addressing Roll et al.'s eighth design principle: communicate the metacognitive learning goals to the students [10].

Completing the set of four tutors is the Interviewer. The Interviewer asks students to answer questions in natural language. The Interviewer serves the metacognitive goal of encouraging students to self-reflect on their process by prompting students to elucidate their decision-making.

3 The Architecture of MeTA

This set of metacognitive tutors for teaching inquiry-driven modeling has been constructed in an experimental architecture titled MeTA, for Metacognitive Tutoring Architecture. At a basic level, the MeTA architecture builds on the characterization of an intelligent agent as a function f that maps a history of percepts P* into an action A; f: P* → A. This section describes MeTA at a software architecture level, consisting of percepts, actions, and mappings between them.

Percepts are defined information the tutor can sense in the learning environment. We have used six categories of percepts for constructing our tutors, including history software and tutor interaction and a current model of student behavior. Actions, in turn, are output complements to the input percepts. Whereas percepts tell tutors for what to look for, actions tell them how to respond. We have used six different categories of actions, including textual feedback, soliciting further information, and altering an underlying model of student behavior. Mappings pair up sets of Percepts with sets of Actions. When every Percept in a given Mapping is observed, the tutor responds with the associated Actions. In many ways, individual tutors can be seen as prioritized lists of Mappings.

4 Initial Deployment & Results from MeTA in MILA

MILA was used in a two-week camp in Summer 2012 with 16 middle school students. The phenomenon that students were charged with explaining was the actual, sudden death of thousands of fish in a nearby lake. To investigate this problem, students took field trips to the lake, participated in physical science and biology exercises, and engaged with MILA in groups of two or three. MILA provided facilities for stating the problem, proposing multiple hypotheses, modeling those hypotheses, consulting static simulations, and researching online hypermedia and data sources. Given that this was the first use of MeTA tutors in a classroom, data gathering and analysis was treated as an exploratory study; the goal, in line with design-based research, was to observe the strengths and weaknesses to better understand how to create effective metacognitive tutors in the future. We found two primary guidelines that are informing our ongoing revisions to the tutoring systems. First, our experience deploying tutors that play multiple functional roles within the software directed our attention to the different ways in which students interact with different roles and types of feedback; this has been similarly touched on elsewhere in research on metacognitive tutoring [3,9]. This has led to the revision of these tutors for new interventions to better differentiate their functional roles and expand the range of types of feedback available. Secondly, we observed the need to address the challenge outlined in Roll et al. 2007 [10] regarding applying one of Anderson et al. 1995's original design guidelines [2] to the metacognitive tutoring domain. This principle – "Facilitate successive approximations of the target skill" – addresses the need to differentiate and address the student's current level of efficacy with the target skill, changing the way in which the skill is addressed as student efficacy changes. Ongoing revisions to the tutors outlined here attempt to equip the system with the ability to infer and address successive approximations of the target skill.

5 Acknowledgments

We are grateful to Kristina Strickland (Little Mill Middle School), Andrew Burch & Chris Thompson (CEISMC), Helen Ellis (Georgia Aquarium), Mary Moerlins (Piedmont Park), Dr. Megan Duffy, Dr. Marc Weissburg, Dr. Jeanette Yen, Nicolas Papin, and Cherish Weiler for their help. This work was partially supported by Proctor & Gamble. We are thankful to Dr. Gautam Biswas for his encouragement and support.

6 References

1. Aleven, V., McLaren, B., Roll, I., & Koedinger, K. (2004). Toward tutoring help seeking. *Intelligent Tutoring Systems, 19*, 105-154.
2. Anderson, J., Corbett, A., Koedinger, K., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *Journal of the Learning Sciences 4.* 167-207.
3. Azevedo, R., Witherspoon, A., Chauncey, A., Burkett, C., & Fike A. (2009). MetaTutor: A MetaCognitive tool for enhancing self-regulated learning. In *Procs. 23rd AAAI Fall Symposium on Cognitive and Metacognitive Educational Systems.*.
4. Bransford, J., Brown, A., & Cocking, R. (2000). *How people learn: Brain, mind, experience, and school*. Washington, D.C.: National Academy Press.
5. Collins, A., Brown, J., & Holum, A. (1991). Cognitive Apprenticeship: Making Thinking Visible. *American Educator 15*(3).
6. Edelson, D., Gordin, D., & Pea, R. (1999). Addressing the challenges of inquiry-based learning through technology and curriculum design. *Journal of the Learning Sciences*, *8*(3-4), 391-450.
7. Goel, A, Rugaber, S, Vattam, S (2009) Structure, behavior & function of complex systems: the SBF modeling language. *AIEDAM,* 23:23-35.
8. Goel, A., Rugaber, S., Joyner, D., Vattam, S., Hmelo-Silver, C., Jordan, R., Sinha, S., Honwad, S., & Eberbach, C. (2013) Learning Functional Models of Complex Systems: A Reflection on the ACT project on Ecosystem Learning in Middle School Science. In *International Handbook on Meta-Cognition and Self-Regulated Learning*, R. Azevedo & V. Aleven (editors), pp. 545-560, Berlin: Springer.
9. Graesser, A., VanLehn, K., Rosé, C., Jordan, P., & Harter, D. (2001). Intelligent tutoring systems with conversational dialogue. *AI Magazine*, *22*(4), 39.
10. Roll, I., Aleven, V., McLaren, B., & Koedinger, K. (2007). Designing for metacognition—applying cognitive tutor principles to the tutoring of help seeking. *Metacognition in Learning 2*(2).
11. Schwarz, D., Chase, C., Chin, D., Oppezzo, M., Kwong, H., Okita, S., Biswas, G., Roscoe, R., Jeong, H., & Wagster, J. (2009) Interactive Metacognition: Monitoring and Regulating a Teachable Agent. In D.Jacker, J. Dunlosky, & A. Graesser (eds.), *Handbook of Metacognition in Education*.
12. Vattam, S., Goel, A., Rugaber, S., Hmelo-Silver, C., Jordan, R., Gray, S, & Sinha, S. (2011) Understanding Complex Natural Systems by Articulating Structure-Behavior-Function Models. *Journal of Educational Technology & Society, 14*(1): 66-81.
13. White, B. & Frederiksen, J. (1998). Inquiry, modeling, and metacognition: Making science accessible to all students. *Cognition and Instruction, 16*(1), 3-117.

Evaluation of a Data Mining Approach to Providing Adaptive Support in an Open-Ended Learning Environment: A Pilot Study

Samad Kardan[1] and Cristina Conati[1]

[1]Department of Computer Science, University of British Columbia
2366 Main Mall, Vancouver, BC, V6T1Z4, Canada

{skardan, conati}@cs.ubc.ca

Abstract. This paper describes the initial evaluation results for providing adaptive support based on effective/detrimental interaction patterns discovered by applying data mining on user interaction data for an Interactive Simulation. Previously, we presented the process of building a classifier user model for the AIspace CSP applet, an open-ended interactive simulation which helps with learning how to solve constraint satisfaction problems. In a later work, we presented a methodology for providing adaptive interventions based on the class association rules that form our classifier user model. In this work, we discuss how to use the generated adaptation rules for delivering adaptive support in the form of hints. The initial qualitative evaluation of the resulting support mechanism, as well as a quantitative evaluation using eye tracking and action logs, show that the interventions were well-received by users.

Keywords: Adaptive Interventions, Interactive Simulations, Eye Tracking

1 Introduction

Interactive Simulations (IS hereafter) are increasingly used as learning tools, where they present an open-ended and exploratory environment to support learning in many different disciplines. These ISs are designed to foster exploratory learning by giving students the opportunity to practically and proactively experiment with concrete examples of concepts and processes they have learned theoretically. However, it has been shown that if the students are left to experiment and explore without any additional support, many will show suboptimal interaction behaviors (e.g., [1]) and may not learn well from this form of interaction (e.g., [2]). These students can benefit from having additional support in the form of scaffolding while interacting with this type of Open-Ended Learning Environments (OELEs) (e.g., [3]). The Constraint Satisfaction Problem (CSP) Applet is one of the collection of interactive tools for learning common Artificial Intelligence algorithms, called AIspace [4]. The CSP applet is an Interactive Simulation designed to help students deepen their understanding of solving constraint satisfaction problems. We intend to add adaptive support to the CSP applet to help students use the applet effectively for learning. Implementing adaptive interventions requires adding two components to an OELE: (1) a **user model** that determines if and when to intervene, with additional information on which interventions are appropriate at the time; and (2) an **intervention mechanism** that delivers different interventions based on the assessment of the student model.

Due to the open-ended nature of the interactions with ISs, providing intelligent support is challenging because many different possible behaviors should be taken into account and most often it is not known a priori which behaviors are effective and which ones are not. All this makes developing a successful intelligent support mechanism time consuming [5]. To address these challenges in a timely and generalizable manner, we employ Educational Data Mining [6] methodologies. Our goal is to find relevant patterns in user interaction data in an IS (e.g. the CSP applet) that leads to different levels of user performance. Then, build a user model based on these patterns and finally, use these patterns to extract adaptation rules for delivering relevant adaptive interventions.

To achieve this goal, first we developed a user modeling framework that utilizes user clustering and class association rules mining to identify relevant user types/behaviors from interface actions [7]. Then, we devised a methodology for using the discovered association rules to generate adaptation rules which are then transformed to adaptive interventions [8]. This paper describes the initial evaluation of adaptive interventions that are implemented following our proposed process.

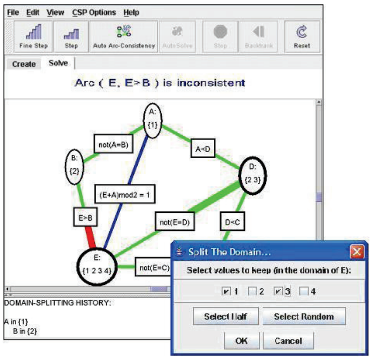
The rest of the paper is organized as follows: First, we briefly describe the CSP applet, the user modeling framework used for extracting user behaviors (i.e., the class association rules), and the methodology for generating adaptation rules based on these behaviors. Then, we discuss the different dimensions for providing interventions based on these adaptation rules. Finally, we present the results of a pilot study with a new version of the CSP applet that implements the proposed support mechanism.

2 The AIspace CSP applet

A CSP consists of a set of variables, variable domains and a set of constraints on legal variable-value assignments. Solving a CSP requires finding an assignment that satisfies all constraints. The CSP applet illustrates the Arc Consistency 3 (AC-3) algorithm for solving CSPs represented as networks of variable nodes and constraint arcs. AC-3 iteratively makes individual arcs consistent by removing variable domain values inconsistent with a given constraint, until all arcs have been considered and the network is consistent. Then, if there remains a variable with more than one domain value, a procedure called domain splitting is applied to that variable in order to split the CSP into disjoint cases so that AC-3 can recursively solve each case.

The CSP applet demonstrates the AC-3 algorithm dynamics via interactive visualizations on graphs using color and highlighting, and graphical state changes are reinforced through textual messages. The applet provides several mechanisms for the interactive execution of the AC-3 algorithm on a set of available CSPs. These mechanisms are accessible through the toolbar, or through direct manipulation of graph elements. The user can perform seven different actions: (1) Fine Step: use the fine step button to see how AC-3 goes through its three basic steps (selecting an arc, testing it for consistency, removing domain values to make the arc consistent); (2) Direct Arc Click: directly click on an arc to apply all these steps at once; (3) Auto AC:

automatically fine step on all arcs one by one using the auto arc consistency button; (4) Stop: pause auto arc consistency; (5) Domain Split: select a variable to split on, and specify a subset of its values for further application of AC-3 (see pop-up box in the bottom right of Fig. 1); (6) Backtrack: recover alternative sub-networks during domain splitting; (7) Reset: return the graph to its initial status.



Arc (E, E>B) is inconsistent

Select values to keep (in the domain of E):

DOMAIN-SPLITTING HISTORY:

A in {1}
B in {2}

Fig. 1. CSP applet with example CSP problem

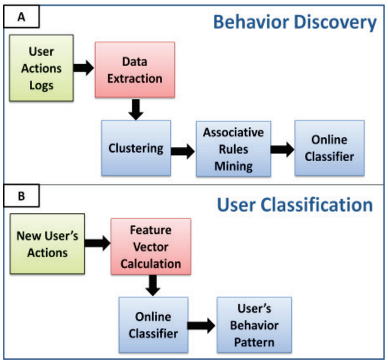


Fig. 2. General User Modeling Approach.

3 Mining Behavior Patterns

In this section we briefly describe the two main phases of our approach to building a classifier user model from interaction data first described in [7]: Behavior Discovery (Fig. 2A) and User Classification (Fig. 2B). In *Behavior Discovery*, raw unlabeled data from interaction logs is preprocessed into feature vectors representing individual users in terms of their interface actions. These vectors are the input to an unsupervised clustering algorithm (i.e., k-means with a modified initialization step, see [7]) that groups them according to their similarities. The resulting clusters represent users who interact similarly with the interface. These clusters are then analyzed to identify if/how they relate to learning. Afterwards, association rule mining is applied on each cluster to extract the common behavior patterns in the form of class association rules for each performance level. A Class Association rule is a rule in the form of X→ *c*, where X is a set of feature-value pairs and *c* is the predicted class label (i.e., the cluster) for the data points where X applies (see Table 1).

Our goal is to use these detected behaviors and information regarding their effectiveness as a guide for intelligent adaptive support during the interaction. Thus, in the *User Classification* phase (Fig. 2B), class association rules extracted in the Behavior Discovery phase are used to build an online classifier user model. This classifier is used to assess the performance of a new user based on her interactions.

In [7], we reported the result of applying our framework on the action logs collected from a study with 65 users using the CSP applet. For this dataset, the Behavior Discovery resulted in two clusters of users that achieved significantly different learning performance levels (high vs. low). We will refer to them as High Learning Gain (HLG) and Low Learning Gain (LLG) groups respectively. Also, the online classifier achieved an accuracy of over 80% in classifying new users as HLG or LLG by observing only the first 25 percent of their interactions.

In addition to assigning a label to the user, the user model also returns the observed rules that caused that classification decision. In [8], we described our proposed methodology for building an intervention mechanism based on the discovered behavior patterns which is briefly described in the next section.

4 Extracting Adaptation rules from Discovered Patterns

The class association rules generated in the Behavior Discovery phase represent the interaction behaviors of LLG and HLG. All of these rules are used in the classifier user model to determine the performance of a new user, and identify a set of behaviors that are either conducive or detrimental to learning. Ideally, one would want to design adaptive interventions that discourage all the detrimental behaviors, and encourage all the good ones. For instance, consider the following rule for the LLG:

Rule4: If Direct Arc Click frequency = Lowest **and** Direct Arc Click Pause Average = Lowest → Cluster LLG

This rule indicates that if the frequency of Direct Arc Click (DAC) action is lower than a threshold (the mechanism to set this threshold is described in [7]) and the average pause time between a DAC and the next action is also lower than a certain threshold then the user is considered a LLG. Here, we want to prevent this from happening and there are two possible interventions (***intervention items*** from now on) that can be delivered to address this rule: (1) Encouraging/enforcing the user to perform DAC more often; (2) Encouraging/enforcing the user to pause longer after DAC actions (possibly thinking about the DAC outcomes).

There may be several rules like the one above that are applicable at a given time. The number of rules, may pose a challenge considering factors such as the cost of implementation and effectiveness of the resulting intervention items, thus filtering the rules is necessary (see [8] for a detailed discussion). For each intervention item, we compute a score calculated as the sum of the weights of the rules which recommend that item within a given cluster (these weights indicate the importance of each rule in classifying a user [7]) and use this as an importance factor for that item. Then we apply a filtering strategy that keeps the most prominent behaviors and ignores the weaker ones while taking the diversity of the intervention items and their cost of implementation into account (see [8] for details). For our current study, we use 6 intervention items as selected by our filtering strategy, highlighted in Table 1.

Table 1. A selection of representative rules for HLG and LLG clusters in the CSP dataset

Rules for HLG cluster: **Rule1:** Direct Arc Click frequency = Highest **Rule5:** Domain Split frequency = Highest **and** Auto AC frequency = Lowest └ **Rule8:** Domain Split frequency = Highest **and** Auto AC frequency = Lowest **and** Fine Step Pause Average = Highest **and** Reset frequency = Lowest
Rules for LLG cluster: **Rule1:** Direct Arc Click Pause Average = Lowest **Rule3:** Direct Arc Click frequency = Lowest

When delivering the implemented interventions to a user, there can be more than one rule satisfied at a certain time leading to multiple items being recommended to that user. If the items are semantically correlated (as determined by the system designer), there is an opportunity to combine two items into one hint. For instance, based on the light blue items in Table 1, a hint can recommend using Direct Arc Click instead of Auto AC, because Direct Arc Click is a finer-grained version of Auto AC, with added user involvement (semantically correlated items have the same color in Table 1). However, non-related items will need separate hint messages and we decided to deliver only one hint at a time to prevent users from possibly getting overwhelmed. Therefore, in each step we choose the intervention item with highest score, calculated similar to above but only for the satisfied rules that recommend that item.

Adaptation rules can be categorized into two main groups, (1) Preventive interventions that discourage bad behavior as detected by the rules for LLG cluster, e.g.: "IF user is classified as a LLG and is using Direct Arc Click very infrequently (less than a threshold), then give a hint to promote this action"; and (2) Prescriptive interventions that encourage the effective behaviors described by the rules for HLG cluster. In this case, we want these rules to be satisfied. This means that if a student labeled as LLG shows any behavior in contrast with these rules then the corresponding intervention will be delivered to her, e.g.: "IF the user label is LLG, then if *Direct Arc Click frequency* is lower than *x* and *Auto AC frequency* is higher than *y* then "prompt user to use Direct Arc Click instead of Auto AC".

The advantage of preventive interventions is that we already know these behaviors result in bad performance so we can confidently prevent users from following such patterns. Prescriptive interventions are less reliable because it is not clear if/how behaviors that were effective for some learners could be beneficial for others.

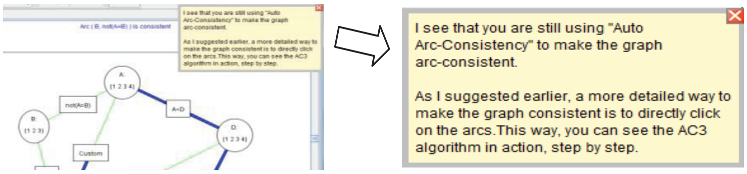
5 Designing adaptive interventions

There are different forms of adaptive interventions that can be used to implement a specific adaptation goal (in our case, helping students use and learn most effectively from the CSP applet). Similar to most of the educational environments that provide adaptive support, we provide explicit advice via textual hints, and provide this advice incrementally. However, our focus on the interface actions when extracting the user interaction behaviors enables us to make interface changes as another way of delivering interventions. Thus, we provide a first level of advice with a textual hint that suggests or discourages a target behavior, followed when needed by a textual hint that reiterates the same advice, accompanied by a related interface adaptation (e.g., highlighting or deactivating relevant interface items).

Delivering adaptive interventions also require deciding whether the interventions should be subtle or forceful. Subtle interventions are in the form of suggestions that can be easily ignored by the user (e.g. a text message shown in a hint box at the corner of the screen). Forceful interventions make the user follow the related advice by reducing or eliminating user's options for the next action (e.g. deactivating all the items on the toolbar to force the user to pause before taking next action).

The current adaptive version of the CSP applet uses the subtle approach. The main drawback of this approach is that the recommendations may not be attended to by users or the user might decide not to follow them. However, this approach has the very desirable advantage of being less intrusive than the forceful approach. Therefore, from a usability point of view, it makes sense to try and see whether subtle adaptive interventions can already significantly improve the effectiveness of the CSP applet.

The detailed procedure of delivering the subtle incremental interventions described above is as follows: (1) for each intervention there is a text message presented in format of a hint that appears in a hint box at the upper left corner of the applet's main panel (level-1 hint). The hint box will blink once, each time a new message is displayed. (2) After receiving the hint, the student is given a time window to change her behavior. (3) If after the time window, the preconditions for that intervention are still satisfied the intervention will be provided again. In this case in addition to a text message, corresponding interface element(s) for that intervention will be highlighted until the user chooses her next action (level-2 hint). Figure 3 shows a level-2 intervention suggesting a decrease in use of *Auto AC* vs. an increase in use of *Direct Arc Click*. In addition to a text message the arcs that can be clicked are also highlighted.



I see that you are still using "Auto Arc-Consistency" to make the graph arc-consistent.

As I suggested earlier, a more detailed way to make the graph consistent is to directly click on the arcs.This way, you can see the AC3 algorithm in action, step by step.

Fig. 3. A hint suggesting the use of Direct Arc Click action with the interface highlights (left); and the content of the hint box (right).

6 Evaluation

We ran a pilot study in a Wizard-of-Oz setting (i.e., experimenter would trigger the interventions based on a set adaptation rules) to evaluate the intervention mechanism described above for three factors: visibility, intrusiveness, and follow rate of the interventions. The data was collected from 6 computer science students. Each participant: (1) studied a textbook chapter on the AC-3 algorithm; (2) wrote a pre-test on the concepts covered in the chapter; (3) used the CSP applet to study two CSPs, while her gaze was tracked with a Tobii T120 eye-tracker; and (4) took a post-test analogous to the pre-test [9]. At the end of the experiment, a qualitative evaluation of interventions was done using a post-hoc questionnaire and a follow-up interview.

Figure 4 summarizes the opinion of our 6 participants about the text hint messages collected by the post-hoc questionnaire. The participants did not find the hint messages intrusive or annoying. They found the messages easy to notice and useful in the process of interaction. Moreover, most of the participants reported following the instructions provided in the hints. The rest of this section will present quantitative results derived from action logs and eye gaze data collected during the interaction.

Regarding visibility of the hints, out of 27 hints provided in total, 25 of them were attended to by the participants. One of two omitted hints was a level-1 hint given to participant 4 (P4), while she did not notice this hint, the subsequent level-2 of the same hint (with interface highlights) managed to grab her attention. The second case was a level-2 hint given to P6, where he decided not to follow a level-1 hint prior to this hint and was given a level 2 hint. In this case, the highlighting reminded him of the recommended action (Direct Arc Click) from the level-1 hint, thus he followed the hint without having to look at the hint box. These two cases, highlight the importance of the 2-level hinting strategy reinforced by interface changes.

Figure 5 illustrates the number of hints shown, attended to and followed by each participant. Out of 27 hints given, 20 were followed by the participants (74% follow rate). Students, who show many detrimental behaviors, will get more hints. Such students are the target group that we want to help learn better from their interaction with the CSP applet. Therefore, P2 and P4 are of especial interest. Both of these participants reported finding the interventions relevant and useful. However, P4 did not follow every hint, and generally only followed the recommendations when repeated in the form of a level-2 hint. This is reflected in her self assessment of how often she followed the hints as well (Table 2).

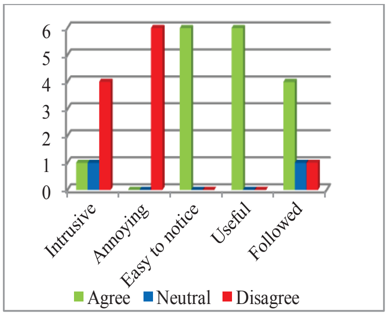


Fig. 4. Reception of the text hints by participants

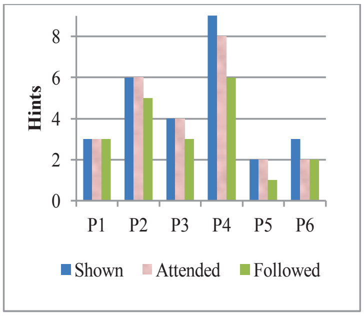


Fig. 5. Number of hints shown, attended and followed for each participant

We also analyzed the average reading time of the hint messages for each participant, overall and for the hints they dismissed/followed (Table 2). We can observe an individual element in reading time between participants which can be further investigated as a guide for user adaptive reaction time for hints. Another trend is that users who received more hints also spent less time reading them. This is expected as these users are the ones with sub-optimal interaction behaviours and this again shows the importance of the 2-level progressive hinting strategy which gets more intrusive the second time a hint is provided.

Table 2. Hint rate, self rated following of hints, and average reading time for each participant

	P1	P2	P3	P4	P5	P6
Followed Hints - Self-rated (1-5)	4	4	4	2	4	3
Avg. Reading Time (ms)	2814	1642	1547	925	2639.5	9460
Avg. Reading Time: Followed (ms)	2814	1530.6	1663	937.5	3464	8975
Avg. Reading Time: Dismissed (ms)	-	2199	1199	887.5	1815	9945
# Hints given	3	6	4	9	2	3

7 Conclusion and future work

In this paper, we presented the final step of the process for adding adaptive interventions to an OELE called AIspace CSP applet. This process started with mining behavior patterns in the form of association rules from a dataset of collected user interface actions [7]. Then, continued with extracting adaptation rules from the discovered behaviors [8]. The final step was to deliver the adaptive interventions defined based on the adaptation rules via an intervention mechanism. We identified the *form* and *forcefulness* of delivering the interventions as two aspects of this step and described our 2-level subtle method of delivering interventions using both text messages and interface changes. The very encouraging initial results of our pilot study regarding reception of the interventions by the users, shows a great potential for the Adaptive version of the CSP applet which provides personalized support. A second pilot study is scheduled to test the user model and the improvements made to the applet based on our findings in the first pilot study. We plan to run a full scale study afterwards.

References

1. Ploetzner, R., Lippitsch, S., Galmbacher, M., Heuer, D., Scherrer, S.: Students' difficulties in learning from dynamic visualisations and how they may be overcome. Computers in Human Behavior. 25, 56–65 (2009).
2. Shute, V.J.: A comparison of learning environments: All that glitters. Computers as cognitive tools. pp. 47–73. Hillsdale, NJ, England: Lawrence Erlbaum Associates, Inc (1993).
3. De Jong, T.: Technological Advances in Inquiry Learning. Science. 312, 532–533 (2006).
4. Amershi, S., Carenini, G., Conati, C., Mackworth, A.K., Poole, D.: Pedagogy and usability in interactive algorithm visualizations: Designing and evaluating CIspace. Interacting with Computers. 20, 64–96 (2008).
5. Cocea, M., Gutierrez-Santos, S., Magoulas, G.D.: Challenges for intelligent support in exploratory learning: the case of ShapeBuilder. Proceedings of the International Workshop on Intelligent Support for Exploratory Environments at ECTEL 2008. , Maastricht, The Netherlands (2008).
6. Romero, C., Ventura, S.: Educational Data Mining: A Review of the State of the Art. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews. 40, 601–618 (2010).
7. Kardan, S., Conati, C.: A Framework for Capturing Distinguishing User Interaction Behaviours in Novel Interfaces. In: Pechenizkiy, M., Calders, T., Conati, C., Ventura, S., Romero, C., and Stamper, J. (eds.) Proceedings of the 4th International Conference on Educational Data Mining. pp. 159–168. , Eindhoven, the Netherlands (2011).
8. Kardan, S., Conati, C.: Providing Adaptive Support in an Exploratory Learning Environment by Mining User Interaction Data. Proceedings of the 5th International Workshop on Intelligent Support for Exploratory Environments (ISEE 2012), in conjunction with the 11th International Conference on Intelligent Tutoring Systems (ITS 2012). , Chania - Greece (2012).
9. Kardan, S., Conati, C.: Exploring Gaze Data for Determining User Learning with an Interactive Simulation. In: Masthoff, J., Mobasher, B., Desmarais, M., and Nkambou, R. (eds.) User Modeling, Adaptation, and Personalization. pp. 126–138. Springer Berlin / Heidelberg (2012).

Adaptive Multi-Agent Architecture to Track Students' Self-Regulated Learning

Babak Khosravifar[1], Roger Azevedo[1], Reza Feyzi-Behnagh[1], Michelle Taub[1], Gautam Biswas[2], and John S. Kinnebrew[2]

[1]McGill University, The Laboratory for the Study of Metacognition and Advanced Learning Technologies, Montreal, Canada
[2]Vanderbilt University, The Teachable Agents Group at Vanderbilt University, Nashville, USA
(babak.khosravifar,roger.azevedo)@mcgill.ca,
(reza.feyzibehnagh,michelle.taub)@mail.mcgill.ca,
(gautam.biswas,john.s.kinnebrew)@vanderbilt.edu

Abstract. Intelligent Tutoring Systems (ITS) can be designed to improve learning and performance through Pedagogical Agents (PAs) that are designed to foster self-regulated learning through interactions and exchange of information with human learners. PAs are intelligent and follow rational behaviors, but to adaptively track students' progress, they need to be systematically and specifically designed. However, in order to follow a common goal, different self-regulatory systems have been designed that use PAs, but fail to provide an adaptive multi-agent architecture which provides such feature that agents adaptively track students' scaffolding. In this paper, we introduce a multi-agent framework designed for an agent-based ITS. We also define the agent architecture, multi-agent framework and communication mechanism.

Keyword. Pedagogical Agents, Self-Regulated Learning, Multi-Agent Systems, Agent Communication Mechanism.

1 Introduction

Increasing adaptivity is being devoted to frameworks involving intelligent components that receive (or search for) data and dynamically update their internal engine to efficiently acquire and integrate information. This adaptivity is becoming a crucial feature in ITSs that provide scaffolding for students to effectively self-regulate their learning. There are various ITSs [1–4, 6], which are used to conduct educational research. But in this paper, we only concentrate on agent-based ITSs [1, 3, 4, 6] where PAs continuously interact with students and objectively provide guidance to facilitate the process of learning and use of effective SRL processes. We concentrate on this category of ITSs because agents are intelligent components that could be equipped with adaptive applications and dynamically track student behaviour, based on the scaffolding they are receiving.

Current ITSs are not entirely adaptive to students' knowledge acquisition during learning in real-time. This may be because in most agent-based ITSs [1, 4, 6], agents are developed to interact with students to facilitate their navigation through parts of the system and provide adaptive scaffolds and feedback to facilitate their learning. This is done using rule-based (predefined) decision maker modules that pick a specific action, which can be either feedback to the students or some sort of communication with the system. The action selection mechanism has been thoroughly defined and enables agents to effectively react to students' progress based on predefined scenarios. In such ITSs, agents generally have a narrower focus on specific performance features/outcomes that illustrate acquisition of knowledge in the target domain.

To address the aforementioned adaptivity problem, PAs need to maintain decision making procedures [5] that continuously interact with the student (in the form of direct interaction and recording the data about that interaction) and dynamically analyze the collected data to update the scaffolding model that the agent builds as it assesses students' progress. By analyzing collected data, agents are able to better interact with students since they are aware of students' detailed work and progress in learning. In this paper, we focus on a multi-agent framework designed for an agent-based ITS that is being designed to analyze a much wider array of student behavior, activities, responses to agents, and performance in order to better understand many aspects of both students' understanding of domain knowledge and underlying self-regulatory abilities.

2 Multi-Agent Architecture

The proposed multi-agent architecture is a simulation environment designed to model and scaffold learners' SRL processes as they learn a biology topic. This environment is focused on further understanding of students' deployment of SRL processes by providing a computer-based learning environment with Pedagogical Agents (PAs) that model and track students' progress while learning complex science topics. In the proposed muti-agent architecture, there are three PAs that directly interact with students:

Peer agent, that interacts the most with the student and obtains basic information (like his/her knowledge level) from the student. In fact, the peer agent is the one that builds the student model and dynamically updates the model with respect to students' activities and deployment of SRL processes;

SRL agent, that tracks students' progress towards using effective SRL processes. This agent is in charge of guiding the student in accomplishing the learning goal and effectively finalizing the process of learning about the complex topic. The SRL agent also provides relative data (computed knowledge level) that influence the peer agent's further interaction with the student.

Science agent, that is in charge of helping and scaffolding the student to understand the science content. This agent informs the other two agents when the student is having difficulties with the content, choosing relevant page sequences, reading the content at an optimal time, and evaluating his/her goals.

The three introduced agents directly interact with students and are known by students as their interactive partners. These agents also interact with each

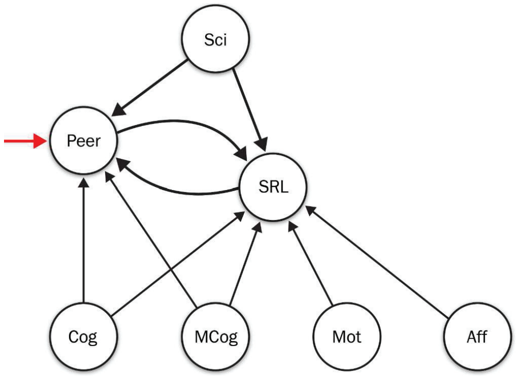


Fig. 1. Multi-agent framework.

other to better guide the student to accomplish the goal of learning about the complex science topic. To adaptively track and model students' scaffolds, there are various data types regarding students' use of SRL processes that need to be collected and analyzed in order to maintain adaptive scaffolding and provide effective guidance to the student. In the proposed framework, we assign four hidden agents, each of which are associated to a category that captures related data, analyses the data and provides relative reports in the form of messages to other involving agents. These massive data is categorized into four groups:
Cognitive agent, that provides details regarding students' learning-related parameters, including their content reading process, highlighting, note taking, and all other cognitive processes;
Metacognitive agent, that provides details regarding students' performance-related parameters, such as scores on various quizzes, accuracy of judgment of learning, and all other metacognitive processes;
Motivational agent, that provides details regarding students' task difficulty, attributions, self-efficacy;
Affective agent, that provides details regarding students' motivations while interacting with the system.

The whole architecture enhances the performance of data collection, and analyzes agents' decision making. Moreover, the multi-agent architecture provides modular functionalities that makes it simpler to test, analyze, and integrate in the system. Figure 1 illustrates the multi-agent architecture together with the involved agents. Hidden agents are rational intelligent components that are capable of analyzing data related to a specific architecture and a pre-defined logic. PAs are rational and are developed with goals related to educational purposes, such as, to optimize learning for students. The core of an agent architecture is its data processing engine that analyses the data that is collected from the surrounding environment and provides an action that best fits its goal-directed purpose. In the proposed architecture, PAs also run data analyses and react to the environment via a selected action by the student. We focus here on the obtained data that help (whether one of the three PAs or the four hidden agents) to analyze and better understand environmental changes, specifically students' decisions and actions.

In the proposed architecture, hidden agents continuously communicate to capture students' activities while interacting with the system and therefore provide accurate information, evidence, and reasoning to the three interactive agents who can then adaptively provide feedback and scaffolds to the students. In the proposed architecture, the main role of these four agents is to collect data regarding cognitive, metacognitive, motivational, and affective SRL processes. These massive data are continuously collected, analyzed and updated to adaptively track their learning progress and adaptations based on the scaffolding they are being provided with.

3 Conclusion

This paper introduces an adaptive multi-agent framework designed for intelligent tutoring systems. This framework could be used in agent-based learning environments where pedagogical agents coordinate with one another to facilitate SRL processes in learners [3]. The main objective is to enable PAs to effectively track students' progress while interacting with the system throughout the learning session. In future research, we intend to propose different mechanisms to develop adaptive multi-agent communication and decision making to represent an optimally efficient learning environment to facilitate the acquisition, internationalization, application, and transfer of self-regulatory processes.

References

1. Azevedo, R., Witherspoon, A., Chauncey, A., Burkett, C., and Fike, A. (2009). MetaTutor: A meta- cognitive tool for enhancing self-regulated learning. In R. Pirrone, R. Azevedo, G. Biswas, (eds.), Proceedings of the AAAI Fall Symposium on Cognitive and Metacognitive Educational Systems (pp. 14-19). Menlo Park: Association for the Advancement of Artificial Intelligence (AAAI) Press.
2. Azevedo, R., Behnagh, R., Duffy, M., Harley, J., and Trevors, G. (2012). Metacognition and self-regulated learning in student-centered leaning environments. In D. Jonassen and S. Land (Eds.), Theoretical foundations of student-center learning environments (2nd ed.)(pp. 171-197). New York: Routledge.
3. Biswas, G., Jeong, H., Kinnebrew, J., Sulcer, B., and Roscoe, R. (2010). Measuring self-regulated learning skills through social interactions in a Teachable Agent Environment. Research and Practice in Technology-Enhanced Learning, 5(2), 123-152.
4. Graesser, A. C., and McNamara, D. S. (2010). Self-regulated learning in learning environments with pedagogical agents that interact in natural language. Educational Psychologist, 45, 234-244.
5. Ross, S., Chaib-draa B., and Pineau J. (2008). Bayesian reinforcement learning in continuous POMDPs with application to robot navigation. International Conference on Robotics and Automation (ICRA), pp. 2845-2851. Jochen Triesch: IEEE.
6. Woolf, B. (2009). Building intelligent interactive tutors: Student-centered strategies for revolutionizing E-Learning. San Francisco, CA: Morgan Kaufmann.

A Differential Temporal Interestingness Measure for Identifying the Learning Behavior Effects of Scaffolding

John S. Kinnebrew, Daniel L.C. Mack, and Gautam Biswas

Department of EECS and ISIS, Vanderbilt University, Nashville TN 37212, USA,
john.s.kinnebrew@vanderbilt.edu

Abstract. Effective design and improvement of scaffolding in complex and open-ended learning environments, requires the ability to assess the effectiveness of a variety of scaffolding options, not only in terms of overall performance and learning, but also in terms of more subtle effects on students' behavior and understanding. In this paper, we present a novel data mining technique that aids the analysis of scaffolding and students' learning behaviors by identifying activity patterns that distinguish groups of students (e.g., groups that received different scaffolding and feedback during an extended, complex learning activities) by differences in both total behavior pattern usage and evolution of pattern usage over time. We demonstrate the utility of this technique through application to student activity data from a recent experiment with the Betty's Brain learning environment and four different scaffolding conditions.

Keywords: learning behaviors, interestingness measure, sequence mining, information gain

1 Introduction

In order to more effectively teach and promote skills required in the modern world of near-ubiquitous computing and internet connectivity, computer-based learning environments have become more complex and open-ended. This complexity also drives a need for dynamic and adaptive scaffolding that can support students in understanding how to employ and learn with these environments and tools. However, in order to effectively design and improve such scaffolding, we must first be able to assess the effectiveness of a variety of scaffolding options, not only in terms of overall performance and learning, but also in terms of more subtle effects on students' behavior and understanding. In this paper, we present a novel data mining technique that aids the analysis of how students' learning behaviors and strategies are employed with differing frequency over the course of learning or problem-solving activities as the result of different scaffolds and feedback that can be provided in a learning environment.

Identifying sequential patterns in learning activity data can be useful for discovering and understanding student learning behaviors. Researchers have applied sequence mining techniques to a variety of educational data in order to better understand learning. The primary sequential pattern mining task is to discover sequential patterns of items that are found in many of the sequences in a given dataset [1, 2]. For example, Perera *et al.* ([3]) use sequential pattern mining to provide mirroring and feedback tools to support effective teamwork among students collaborating on software development using an open source professional development environment called TRAC. Other researchers have also employed sequential pattern mining to identify differences among student groups or generate student models to customize learning to individual students [4–6].

Once these behavior patterns are mined, researchers must interpret and analyze the resulting patterns to identify a relevant subset of important patterns that provide a basis for generating actionable insights about how students learn, solve problems, and interact with the environment. Researchers have developed a variety of measures to utilize properties other than the default of pattern frequency to rank mined patterns [7]. These measures are often referred to as "interestingness measures" and have been applied data mining tasks like sequence mining and association rule mining [8]. To better analyze student learning and behavior, interestingness measures have been used for tasks such as ranking mined association rules (e.g., [9]).

Investigation of the frequency with which a pattern occurs over time can reveal additional information for pattern interpretation and may help identify more important patterns, which occur only at certain times or become more/less frequent, rather than patterns with frequent, but uniform, occurrence over time. In this paper, we present a novel approach, combining sequence mining and an information-theoretic measure for ranking behavior patterns that distinguish groups of sequences (e.g., groups of students in different experimental conditions) by differences in both total pattern usage and the evolution of pattern usage over time. To effectively analyze these patterns and quickly identify trends in the evolution of pattern usage, we employ a related visualization in the form of heat maps.

2 Identifying Interesting Differences in Pattern Usage

In this section, we present the Differential Temporal Interestingness of Patterns in Sequences (D-TIPS) technique, and its novel interestingness measure, for identifying and visualizing patterns that are employed differentially over time among groups of students (e.g., groups that receive different scaffolding in an open-ended learning environment). The first step in analyzing learning activity sequences is to define and extract the actions that make up those sequences from interaction traces logged by the environment. The definition of actions in these sequences for Betty's Brain data is discussed further in Section 3. Given a set of sequences corresponding to the series of actions performed by each student, the D-TIPS technique consists of four primary steps:

1. Generate candidate patterns that are common to the majority of students in at least one group by combining the sets of patterns identified through ap-

plying sequential pattern mining separately to each group's learning activity sequences (with a frequency threshold of 50%).

2. Calculate a temporal footprint for each candidate pattern by mapping it back to locations where it occurs in the activity sequences. Specifically, each sequence is divided into n consecutive slices, such that each contains $\frac{100}{n}\%$ of the student's actions in the full sequence, where n is the chosen number of bins defining the temporal granularity of the comparisons. Corresponding slices for a group (e.g., the first slice from each sequence in the group, the second slice from each, and so on) are then grouped into bins and each action in the slices is marked to indicate whether or not it is the beginning of a pattern match in its original sequence. This set of binned and marked actions defines the temporal footprint of the pattern for the group.
3. Provide a ranking of the candidate patterns using an information-theoretic interestingness measure (described in more detail below) applied to the temporal footprint of each pattern.
4. For the highly-ranked patterns, visualize their temporal footprints using heat maps to identify differences in usage trends and spikes across student groups. Specifically, we employ a two-dimensional heat map where the y-axis is student group and the x-axis is time discretized by temporal bin. In a single row (i.e., for a specific student group), each cell's count is the percentage of total pattern occurrence (with respect to the student group) within the corresponding temporal bin. The use of *percentages* of pattern occurrence allows analysis of temporal variation normalized by the total frequency of the pattern in the group, which will tend to highlight different temporal trends in pattern usage across groups, even when total pattern occurrence differs significantly among groups.

In order to identify more interesting patterns by their difference in temporal usage across groups in step 3, the D-TIPS interestingness measure applies information gain (IG) with respect to pattern occurrence across the groups in each of the n corresponding bins of their temporal footprints. Information gain is defined as the difference in expected information entropy [10] between one state and another state where some additional information is known (e.g., the difference between a set of data points considered as a homogeneous group versus one split into multiple groups based on the value of some other feature or attribute). Information entropy is the amount of expected uncertainty found in a random variable, X, whose value can be called the *class* of the data point. IG when used in classifiers, such as decision trees, is applied to a dataset where each data point has multiple features in addition to its class. The IG of a given feature is then the reduction in expected uncertainty about the correct class of a data point when its feature value is known, or conversely the increase in information about the class of the data point. IG is calculated as the difference between the information entropy of the data without knowledge of the feature values and the conditional information entropy when the feature values are known.

Information gain is leveraged in classifiers to determine which features are most discriminatory because they provide the least amount of uncertainty among classes in the data. D-TIPS applies information gain to determine which patterns are the most interesting because knowledge of their occurrence and temporal location provides the least amount of uncertainty among the student groups. In D-TIPS, each action/data-point's class is its group, and the feature of each data point, for a given pattern, is the combination of whether the action begins an occurrence of the pattern *and* the number of the bin in which the action occurred. This information-theoretic definition of the D-TIPS measure provides two important properties: 1) given two patterns with the same total occurrences for corresponding groups, the pattern with the greater discrimination of groups by *differences in temporal location/bin among groups* will have a higher rank, and 2) given two patterns with the same relative temporal behaviors (i.e., the same proportion of total pattern occurrence in each bin) for corresponding groups, the pattern with the greater discrimination of groups by *differences in total occurrence among groups* will have a higher rank.

Therefore, the D-TIPS measure provides a way of recognizing differences among groups both by total pattern occurrence and by temporal behavior (e.g., decreasing usage versus increasing usage, or spikes in different bins). Further, when the same differences across groups occur for two patterns, the pattern with higher overall frequency will have the higher rank. Thus, D-TIPS tends to emphasize patterns with large relative differences among groups (by total occurrence and/or temporal behavior) even when they are not especially frequent in the overall dataset, while also emphasizing patterns with more moderate differences among groups when the frequency of the pattern in the overall dataset is high. Conversely, D-TIPS tends to deemphasize patterns that are homogeneous across groups (by both relative occurrence and temporal behavior) or that are especially rare in all groups.

3 Betty's Brain Data

The data we employ in the analysis in Section 4 consists of student interaction trace from the Betty's Brain [11] learning environment. In Betty's Brain, students read about a science process and teach a virtual agent about it by building a causal map. They are supported in this process by a mentor agent, who provides feedback and support for their learning activities. The data analyzed here was obtained in a recent study with 68 7^{th}-grade students taught by the same teacher in a middle Tennessee school. At the beginning of the study, students were introduced to the science topic (global climate change) during regular classroom instruction, provided an overview of causal relations and concept maps, and given hands-on training with the system. For the next four 60-minute class periods, students taught their agent about climate change and received feedback on content and learning strategies from the mentor agent.

The study tested the effectiveness of two support modules designed to scaffold students' understanding of cognitive and metacognitive processes important for success in Betty's Brain (details provided in [12]). The *knowledge construction* (KC) support module scaffolded students' understanding and suggested strategies on how to construct knowledge by identifying causal relations in the resources, and the *monitoring* (Mon) support module scaffolded students understanding and suggested strategies on how to monitor Betty's progress by using the quiz results to identify correct and incorrect causal links on Betty's map. Participants were divided into a control and three treatment groups. The knowledge construction (KC) group used a version of Betty's Brain that included the KC support module and a causal link tutorial that they could access at any time and were prompted to enter when the mentor determined they were having difficulty identifying causal links in the resources. The monitoring (Mon) group used a version of Betty's Brain that included the Mon support module and a tutorial about employing link annotations to keep track of links shown to be correct by quizzes. The full (Full) group used a version of Betty's Brain that included both support modules and tutorials. Finally, the control (Con) group used a version that included neither the tutorials nor the support modules.

In Betty's Brain, the students' learning and teaching tasks were organized around seven activities: (1) reading resource pages to gain information, (2) adding or removing causal links in the map to organize and teach causal information to Betty, (3) querying Betty to determine her understanding of the domain based on the causal map, (4) having Betty take quizzes that are generated and graded by the mentor to assess her current understanding and the correctness of links in the map, (5) asking Betty for explanations of which links she used to answer questions on the quiz or queries, (6) taking notes for later reference, and (7) annotating links to keep track of their correctness determined by quizzes and reading. Actions were further distinguished by context details, which for this analysis were the correctness of a link being edited and whether an action involved the same subtopic of the domain as at least one of the previous two actions. The definition of actions in Betty's Brain learning activity sequences are discussed further in [13].

4 Results

To illustrate and characterize the performance of the D-TIPS technique on educational data, we present selected results from its application to student learning activity data in the Betty's Brain classroom study described in Section 3. The D-TIPS analysis identified 560 activity patterns that occurred in at least half of the students in one or more of the four experimental conditions. Given the limited number of students in each condition, we chose to bin pattern occurrence values into fifths of the activity sequences for a broad analysis of their usage evolution over time. Table 1 presents 3 of the top 30 most differentially-interesting patterns identified by D-TIPS across the four scaffolding conditions. For comparison, the average occurrences per student and ranking by that value is also presented. Over half (18) of the 30 analyzed D-TIPS patterns had a rank past 50th by occurrence, with 13 of them ranking beyond 100th, indicating that they would be unlikely to have been considered without D-TIPS.

Table 1. Selected Patterns with D-TIPS and Occurrence Rankings

Pattern	D-TIPS Rank	Occurrence Rank	Avg Occurrence
[Quiz]	3	2	21.8
[Read] → [Note]	18	100	1.7
[Read] → [Read] → [Remove Link^{-}]	29	137	1.4

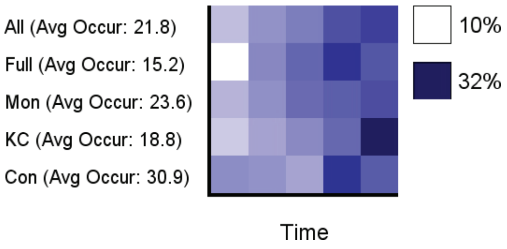


Fig. 1. [Quiz]

The first pattern in Table 1 illustrates a single action pattern that was ranked very high by both D-TIPS and overall occurrence. While individual student actions are often less interesting than longer patterns, they are still important to consider, especially when they also illustrate a tendency to be employed differentially across groups and over time. Figure 1 shows that all groups tended to use quizzes more frequently later in their work on the system. Since students' causal maps grew over time, monitoring and correction of the maps were more important later in their learning activities. There were some differences in usage trends over time among the different conditions, such as the steeper increasing trend for the KC and Full groups than the Monitoring group and the earlier peak in usage for the Full and Control groups. However, the overall occurrence by conditions differed markedly, with the Control group performing far more quiz actions than the others, and the Monitoring group performing more quiz actions than the KC and Full groups. While the Monitoring group's use of the quiz was expected to be high due to the focused monitoring support that relied heavily on the quiz, it is surprising that the Control group had the highest quiz usage. This might indicate that without either KC or monitoring support, the Control group struggled more and fell back on guessing and checking (with the quiz) strategies.

Figure 2 illustrates a knowledge construction behavior of reading and taking notes that was ranked highly by D-TIPS. Another difference among the groups, which added to the interestingness of this pattern under the D-TIPS analysis, is that the Control group tended to perform reading followed by note-taking primarily in the last fifth of their activities, as opposed to the first two fifths for the other groups. However, further analysis of the data attributed this primarily to only two of the Control group students, although the reason for this aberration is still unclear.

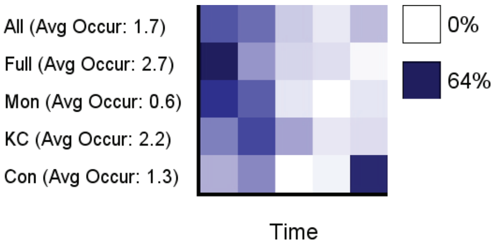


Fig. 2. [Read] → [Note]

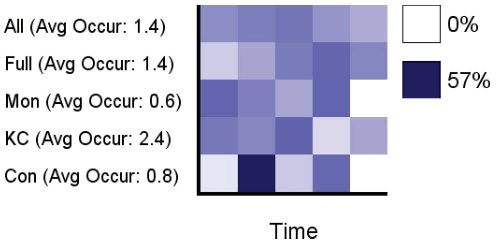


Fig. 3. [Read] → [Read] → [Remove Link$^-$]

The pattern illustrated in Figure 3 involves a sequence of (two) reading actions followed by removing an incorrect link. While there was no consistent temporal trend in the usage of this pattern, the Monitoring and Control groups exhibited this pattern less than once per student, while the KC group averaged 2.4 times per student. Although ranked lower by D-TIPS at 45th, the sub-pattern of a single read action followed by removing an incorrect link illustrates the same differences. This suggests that students with the KC feedback, relied more heavily on reading to identify incorrect links than either the Control and Monitoring groups, possibly because the Control group struggled more in general and the support in the Monitoring group focused students more on the use of quizzes to identify incorrect links.

5 Conclusion

While identification of high-frequency patterns is undoubtedly useful, finding patterns that have differing usage over time across a set of student groups is also important for analyzing the effects of scaffolding. In this paper, we presented the D-TIPS technique, which identifies patterns that differ in their usage among student groups by either total (group) occurrence or temporal behavior, even when they are not especially frequent in the overall dataset. Results from the use of this technique to mine Betty's Brain data illustrated the potential benefits and helped characterize differences between D-TIPS and a baseline occurrence ranking. D-TIPS identified patterns that illustrated potentially important differences in learning behavior among different scaffolding conditions that would have probably been overlooked by considering only pattern frequency. Future work will include autonomous identification of an effective number of bins for splitting a given set of activity sequences, as well as methods to individually characterize student groups by the patterns identified in D-TIPS.

6 Acknowledgments

This work has been supported by NSF-IIS Award #0904387 and IES Award #R305A120186.

References

1. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proceedings of the Eleventh IEEE International Conference on Data Engineering (ICDE). (1995) 3–14
2. Zaki, M.: Sequence mining in categorical domains: incorporating constraints. In: Proceedings of the ninth international conference on Information and knowledge management, ACM (2000) 422–429
3. Perera, D., Kay, J., Koprinska, I., Yacef, K., Zaïane, O.: Clustering and sequential pattern mining of online collaborative learning data. IEEE Transactions on Knowledge and Data Engineering **21**(6) (2009) 759–772
4. Amershi, S., Conati, C.: Combining unsupervised and supervised classification to build user models for exploratory learning environments. Journal of Educational Data Mining **1**(1) (2009) 18–71
5. Martinez, R., Yacef, K., Kay, J., Al-Qaraghuli, A., Kharrufa, A.: Analysing frequent sequential patterns of collaborative learning activity around an interactive tabletop. In: Proceedings of the Fourth International Conference on Educational Data Mining, Eindhoven, Netherlands (2011)
6. Su, J.M., Tseng, S.S., Wang, W., Weng, J.F., Yang, J., Tsai, W.N.: Learning portfolio analysis and mining for scorm compliant environment. Journal of Educational Technology and Society **9**(1) (2006) 262–275
7. Geng, L., Hamilton, H.J.: Interestingness measures for data mining: A survey. ACM Computing Surveys (CSUR) **38**(3) (2006) 9
8. Zhang, Y., Zhang, L., Nie, G., Shi, Y.: A survey of interestingness measures for association rules. In: Business Intelligence and Financial Engineering, 2009. BIFE'09. International Conference on, IEEE (2009) 460–463
9. Merceron, A., Yacef, K.: Interestingness measures for association rules in educational data. Educational Data Mining 2008 (2008) 57
10. Renyi, A.: On measures of entropy and information. In: Fourth Berkeley Symposium on Mathematical Statistics and Probability. (1961) 547–561
11. Biswas, G., Leelawong, K., Schwartz, D., Vye, N., Vanderbilt, T.: Learning by teaching: A new agent paradigm for educational software. Applied Artificial Intelligence **19**(3) (2005) 363–392
12. Biswas, G., Kinnebrew, J.S., Segedy, J.R.: Analyzing students' metacognitive strategies in open-ended learning environments. In: Proceedings of the 35th annual meeting of the Cognitive Science Society, Berlin, Germany (August 2013)
13. Kinnebrew, J.S., Loretz, K.M., Biswas, G.: A contextualized, differential sequence mining method to derive students' learning behavior patterns. Journal of Educational Data Mining (In Press, 2013)

Process and Outcome Benefits for Orienting Students to Analyze and Reflect on Available Data in Productive Failure Activities

Ido Roll, Natasha G. Holmes, James Day, Anthony H.K. Park, and D.A. Bonn

University of British Columbia, 6224 Agricultural Road, Vancouver, BC V6T 1Z1, Canada
{ido,nholmes,jday,bonn}@phas.ubc.ca

Abstract. Invention activities are Productive Failure activities in which students attempt to invent methods that capture deep properties of given data before being taught expert solutions. The current study evaluates the effect of scaffolding on the invention processes and outcomes, given that students are not expected to succeed in their inquiry and that all students receive subsequent instruction. Two Invention activities related to data analysis concepts were given to 130 undergraduate students in a first-year physics lab course using an interactive learning environment. Students in the Guided Invention condition were given prompts to analyze given data prior to inventing and reflect on their methods after inventing them. These students outperformed Unguided Invention students on delayed measures of transfer, but not on measures of conceptual or procedural knowledge. In addition, Guided Invention students were more likely to invent multiple methods, suggesting that they used better self-regulated learning strategies.

Keywords: Invention activities, productive failure, scaffolding, interactive learning environments, transfer.

1 Introduction

Invention activities are activities in which students generate solutions to novel problems prior to receiving instruction on the same topics. For example, students may be asked to generate methods that capture the variability of given data sets prior to being taught about mean deviation [1-3]. Invention activities facilitate Productive Failure in that students commonly fail to generate valid methods in these activities [4-5]. For example, students may use range or count the number of different values as a measure of variability, ignoring distribution and number of data points. However, the failure is often productive as students learn from the subsequent instruction and practice better than students who receive only instruction and practice, controlling for overall time on task [1,3-6].

Unlike other forms of Productive Failure, in Invention activities students are given carefully designed sets of data, called *contrasting cases*, to invent mathematical methods that capture deep properties of data [7-8]. For example, the contrasting cases in Figure 1 are given to students when asked to create a method for calculating a weighted average. The contrast between Carpenters A and C helps students notice and encode the roles of spread and magnitude. The contrast between A and D helps students notice the role of sample size.

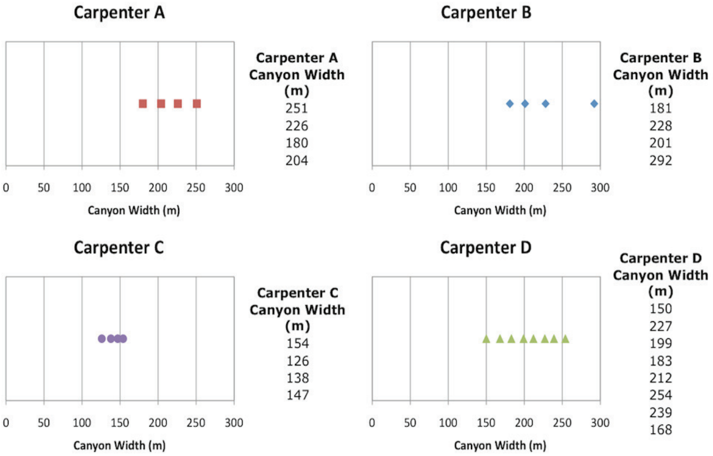


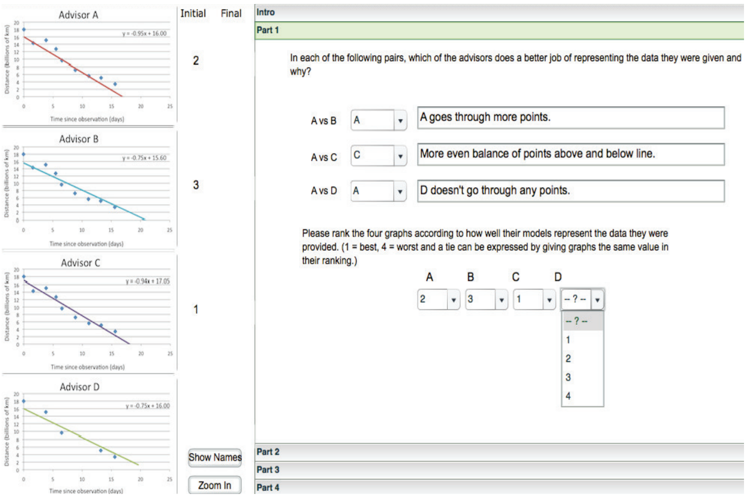
Figure 1 Contrasting cases emphasize the roles of magnitude, distribution, and sample-size in determining weighted average.

The invention process resembles an inquiry process in that students attempt to discover the underlying structure of data [9]. Thus, in the absence of additional support, it is of no surprise that students rarely invent valid methods. However, as described earlier, the invention process improves subsequent learning even in the absence of successful invention [1,2,6]. This raises an interesting question, which we address in this paper: Should the invention process be supported? One hypothesis suggests that supporting invention may lead to improved learning, as students may invent better methods. However, an alternative hypothesis suggests that failure is necessary for learning [10]. Thus, supporting students during their invention process may, in fact, hinder learning.

Scaffolding Invention Activities

One common form of support is scaffolding [11]. Specifically, scaffolding the inquiry process was shown to improve learning in discovery learning [12-13]. Within the context of Invention activities, similar scaffolding was shown to improve the invention process and its outcomes [3]. Within the scope of this study, we chose to focus on scaffolding two key phases that bracket the invention process: orientation and reflection.

Orientation. Invention Activities constrain the inquiry process by offering students contrasting cases to work with. However, simply having the contrasting cases may not be enough. We have previously found that many students working with Invention activities do not engage with the available contrasts when developing their methods [3]. Thus, following a prescriptive cognitive task analysis, we developed and validated prompts that help students orient themselves to the given data. This is done by instructing students to make pairwise comparisons between the contrasting cases with regard to the target concept. For example, students would be asked to compare carpenters A and D in figure 1 to determine which one did a better job of measuring the width of a bridge, see Figure 2. Since the two cases have roughly the same average and spread, students are confronted with the issue of sample size and need to determine whether and how the number of measurements may factor into the problem.



Initial: A 2, B 3, C 1

In each of the following pairs, which of the advisors does a better job of representing the data they were given and why?

A vs B	A	A goes through more points.
A vs C	C	More even balance of points above and below line.
A vs D	A	D doesn't go through any points.

Please rank the four graphs according to how well their models represent the data they were provided. (1 = best, 4 = worst and a tie can be expressed by giving graphs the same value in their ranking.)

A	B	C	D
2	3	1	-- ? --

-- ? --
1
2
3
4

Figure 2. Ranking pairwise contrasts in the orientation scaffold.

Reflecting on the invented method. A second process that we chose to focus on is evaluation and reflection. In addition to being a key process in the scientific toolbox, the process of evaluation is beneficial, as it requires students to self-explain their correct or incorrect reasoning. In the context of Invention activities, once students develop their methods, the scaffolding asks them to explain how their invented methods take into account what they have learned during the pairwise comparisons. Students then apply their invented method to the contrasting cases, and then are asked to evaluate their method by comparing these results to their qualitative rankings as identified by them intuitively in the orientation phase.

Scaffolding students' orientation and reflection processes was found to improve students' invention behaviours and their invented methods on paper [3]. However, we are yet to evaluate the effect of the scaffolding on students' learning gains. The current study evaluates the effect of scaffolding during Invention activities on learning in two ways. First, we evaluate whether scaffolding improves the invention process itself. Given that evaluation and iteration are important inquiry skills, and that multiple invented methods are often associated with better learning in Productive Failure tasks [5], we evaluate the invention process by measuring the likelihood that groups invent more than a single solution. Second, we evaluate the effect of scaffolding on learning outcomes from the overall invention-instruction-practice process. We do so by comparing pre-to-post gains. Notably, these scaffold are static, unlike the view of scaffolding as an adaptive, negotiated process [14]. Understand when students require scaffolding in Productive Failure, and how to detect that using a student model, is outside the scope of the current work.

Method

We compared the Invention activities with and without scaffolding using a pre-to-post design. The *in-vivo* study took place in a first-year physics laboratory course at the University of British Columbia. 130 first-year students from four sections of the course participated in the study. The study was spread across a four-month term with the pre-test and two Invention activities given in three subsequent weeks at the beginning of the term. The final post-test was delivered at the end of the term, roughly two months after students had finished the second invention activity.

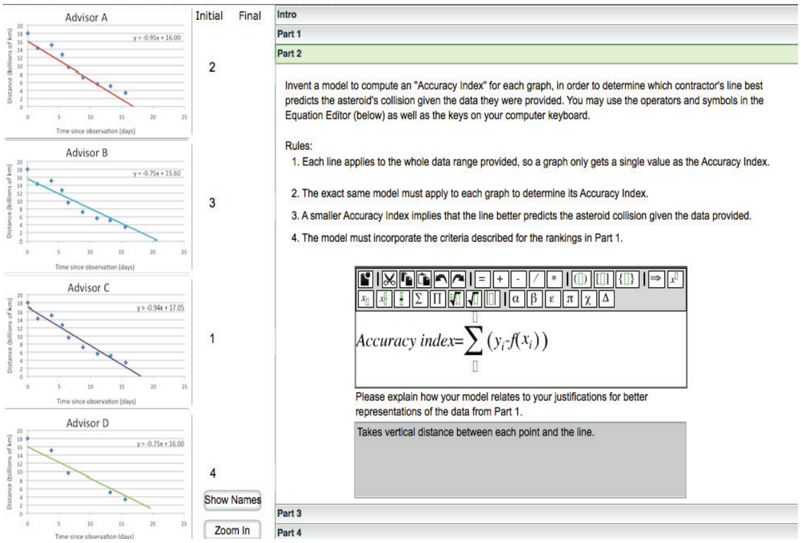
Students were randomly assigned to two groups, and different groups were assembled for the two activities. Students in the Unguided Invention (UI) condition worked with a convention invention activity, as defined in [1.2] (n = 65). Students in the Guided invention (GI) condition received the additional scaffolding, as described below (n = 65). Students were given approximately 30 minutes to work on the Invention activities. Each activity was followed by a short lecture on the target domain from the course instructor, which included a group discussion to direct students' attention to the important features of the data. Following the direct instruction, students worked on scientific experiments for roughly two more hours. These experiments provided opportunities for students to practice applying the expert solution from the Invention activities. Topics from the Invention activities were revisited or built on in subsequent weeks.

All students worked on the Invention activities using a dedicated interactive learning environment, the Invention Support Environment (ISE) [15]. Figure 3 shows the interface of ISE for the second activity used in this study, which focuses on evaluating goodness of fit for linear trendlines. The majority of the screen estates are dedicated to an accordion that breaks down the invention process:

- Introduction: background story and task
- Part 1: orientation. I this phase students analyze the contrasting cases qualitatively (available to GI students only).
- Part 2: generation. In this phase students invent a mathematical method to capture the deep property of the data. This is done using an equation editor (shown in Figure 3).
- Part 3: Students were guided to apply their method using a calculator or a spreadsheet software (e.g., MS Excel), and report back their values.
- Part 4: Students were asked to evaluate their methods based on their qualitative ranking (GI condition only).

The left side of the screen presents the contrasting cases to students. These stay available throughout the process. Students can zoom in on the contrasting cases and see the raw data by clicking on the Zoom In button. The centre of the screen shows students their initial and final ranking, when these are available (GI condition only).

The ISE is a skeleton that can deliver a variety of invention activities that share the same structure. It is used regularly by instructors in this course to deliver roughly 5-6 activities per term. A current version of ISE also includes instruction and opportunities for practice within the environment. Authoring new problems in ISE requires designers to give the text and data, but not to author new behaviours, as these are already built into ISE. ISE was built using the Cognitive Tutor Authoring Tools (CTAT) [16].



Invent a model to compute an "Accuracy Index" for each graph, in order to determine which contractor's line best predicts the asteroid's collision given the data they were provided. You may use the operators and symbols in the Equation Editor (below) as well as the keys on your computer keyboard.

Rules:

1. Each line applies to the whole data range provided, so a graph only gets a single value as the Accuracy Index.
2. The exact same model must apply to each graph to determine its Accuracy Index.
3. A smaller Accuracy Index implies that the line better predicts the asteroid collision given the data provided.
4. The model must incorporate the criteria described for the rankings in Part 1.

$$Accuracy\ index = \sum (y_i - f(x_i))$$

Please explain how your model relates to your justifications for better representations of the data from Part 1.

Takes vertical distance between each point and the line.

Figure 3: The Invention Support Environment

The two conditions differed with regard to support that students received before and after inventing their methods. The scaffolding that was given to students in the GI condition was modeled after the paper scaffoldings that were used in [3]. These scaffolding were designed to promote expert scientific behaviours that were identified in a prescriptive cognitive task analysis using similar invention activities:

The goal of the Orientation prompts was to get students familiar with the data prior to beginning to invent. Students were asked to compare pairs of contrasting cases and rank these according to the target feature. Students were then asked to briefly explain each of their rankings.

To encourage students to reflect on their invented methods, students were explicitly asked to self-explain their invented methods, referring back to their pairwise rankings. In addition, students were explicitly asked to evaluate their methods by comparing the results of their calculated values with their initial ranking during the orientation.

It should be noted that while the UI group did not have explicit prompts to perform these particular steps, they still had the opportunity to engage in them spontaneously. For example, the implementation process often leads naturally to reflection, as students recognize the shortcomings of their formulas, especially if the students spontaneously analyzed the contrasting cases first. Thus, the main difference between the conditions is the explicit prompting to carry out and reflect on each of the key stages. Table 1 summarizes the differences between conditions. Snapshots of the entire process can be found in Appendix B.

The pre- and post- tests included three types of questions on both invention topics. Procedural items asked students to calculate numeric answers by applying the formulas. Conceptual items asked students to apply the concepts without calculation to demonstrate understanding of the basic principles of the domains. Transfer items provided students with equations that were deliberately varied from the domain formulas and asked students to evaluate whether the formulas were reasonable ways to accomplish the same task. This requires a deep understanding of the deep features of the domain and their mathematical expressions in the equations [17]. Each type of assessment had two items, one on each topic.

Results

There was no effect for condition on pre-test: *t*(127) = 0.18, *p* = 0.856 (see Table 1). A paired t-test found significant learning from pre-test (*M* = 0.47, *SD* = 0.24) to post-test (*M* = 0.61, *SD* = 21) on items that were shared by both tests: *t*(129) = 5.75; *p* < .0001.

Overall, 111 pairs of students worked on the two activities (56 pairs on the first activity and 55 pairs on the second). A logistic regression model found that groups in the GI condition were significantly more likely to create multiple methods, controlling for task, GI = 51% UI = 38%; *B* = 1.13, *SE(B)* = 0.56 e^B = 3.091, Z = 4.02, *p* = 0.045. The odds ratio (e^B) suggest that the odds to invent multiple methods is three times as high for GI students compared with UI students.

Table 1. Mean (SD) pre- and post-test scores on procedural, conceptual, and transfer items.

Item Type	Unguided Invention	Guided Invention
Pretest:	28% (31%)	33% (32%)
Posttest:		
- Procedural	46% (31%)	47% (28%)
- Conceptual	75% (28%)	74% (32%)
- Transfer	21% (29%)	33% (35%) *

* p < 0.05

An ANCOVA evaluating the effect of scaffolding on learning found no significant effect for condition on procedural, *F*(2,127) = 0.02, *p* = 0.882; or conceptual knowledge, *F*(2,127) = 0.09, *p* = 0.761. However, condition had a significant effect on transfer items, GI: M = 0.33, SD = 0.35; UI: M = 0.21, SD = 0.29: *F*(2,127) = 4.81; *p* = 0.030.

Discussion and Summary

The results presented above show that adding scaffolding to the invention process led to a higher rate of multiple methods during the invention process and to increased gains on a measure of transfer two months after the initial learning period. The scaffold had no effect on procedural and conceptual items. This is not surprising since the invention process itself usually has no benefits for these items compared with direct instruction and practice alone [1,2,17]. Thus, modifying the invention process similarly has no effect on performance on these items.

One key question to be answered is how the scaffolding resulted in the observed improvements. One likely answer suggests a two-fold process. By requiring students to compare pairs of contrasting cases, students notice more features, thus gaining a fuller understanding of the target domain. Using reflection prompts, the scaffolding improves students' meta-knowledge in that it highlights what is known (features) versus what is yet to be learned (the integrated method). Thus, orientation and reflection prompts help students obtain a fuller understanding of the domain, but not necessarily of any specific method. This may explain the observed effect on transfer, but not other, items.

The study further demonstrates that Productive Failure works not simply because support should be delayed. Instead, it is the transmission of domain knowledge that should be withheld, while other forms of support may be beneficial for learning even using the Productive Failure paradigm [6].

The study has several limitations. Most notably, due to the dynamic allocation of students to groups, we did not directly evaluate the relationship between quality of invention and quality of learning. Future work will have to address this issue, as well as focus on topics other than data analysis.

Notably, adding guidance during Invention activities helps learning even though students commonly fail to invent the expert solutions. Thus, not only that the failure to invent is, indeed, productive, but also, some failures are more productive than others. This study demonstrates how engaging students with good scientific practices helps them achieve a more productive failure.

Acknowledgements

This work was supported by the Pittsburgh Science of Learning Center, which is funded by the National Science Foundation, award number (#SBE-0836012), and by the University of British Columbia through the Carl Wieman Science Education Initiative.

References

1. Roll, I., Aleven, V., & Koedinger, K. R. (2009). Helping students know 'further'-increasing the flexibility of students' knowledge using symbolic invention tasks. In Proceedings of the 31st annual conference of the cognitive science society (pp. 1169-74).
2. Schwartz, D. L., & Martin, T. (2004). Inventing to prepare for future learning: the hidden efficiency of encouraging original student production in statistics instruction. Cognition and Instruction, 22(2), 129–184. Doi:10.1207/s1532690xci2202_1
3. Roll, I., Holmes, N., Day, J., & Bonn, D. (2012). Evaluating metacognitive scaffolding in guided Invention activities. *Instructional Science*, *40*(4), 691–710. doi:10.1007/s11251-012-9208-7
4. Kapur, M. (2008). Productive failure. *Cognition and Instruction*, *26*(3), 379–424. doi:10.1080/07370000802212669
5. Kapur, M., & Bielaczyc, K. (2012). Designing for productive failure. Journal of the Learning Sciences, 21(1), 45–83. doi:10.1080/10508406.2011.591717
6. Westermann, K., & Rummel, N. (2012). Delaying instruction: evidence from a study in a university relearning setting. *Instructional Science*, *40*(4), 673–689. doi:10.1007/s11251-012-9207-8
7. Schwartz, D. L., Chase, C. C., Oppezzo, M. A., & Chin, D. B. (2011). Practicing versus inventing with contrasting cases: The effects of telling first on learning and transfer. *Journal of Educational Psychology*, *103*(4), 759–775.
8. Roll, I., Aleven, V., & Koedinger, K. R. (2010). The invention lab: Using a hybrid of model tracing and constraint- based modeling to offer intelligent support in inquiry environments. In V. Aleven, J. Kay, & J. Mostow (Eds.), Proceedings of the 10th International Conference on Intelligent Tutoring Systems (pp. 115-24). Berlin: Springer Verlag.
9. de Jong, T., & van Joolingen, W. R. (1998). Scientific discovery learning with computer simulations of conceptual domains. *Review of Educational Research*, *68*(2), 179–201. doi:10.3102/00346543068002179
10. VanLehn, K. (1988). *Toward a theory of impasse-driven learning* (pp. 19-41). Springer US.
11. Koedinger, K. R., & Aleven, V. (2007). Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review*, *19*(3), 239-264.
12. de Jong, T. (2006). Scaffolds for scientific discovery learning. In J. Elen, R. E. Clark, & J. Lowyck (Eds.), *Handling complexity in learning environments: Theory and research* (pp. 107–128). Howard House: Emerald Group Publishing.
13. Hmelo-Silver, C. E., Duncan, R. G., & Chinn, C. A. (2007). Scaffolding and Achievement in Problem-Based and Inquiry Learning: A Response to Kirschner, Sweller, and Clark (2006). *Educational Psychologist*, *42*(2), 99–107. doi:10.1080/00461520701263368
14. Wood, D., Bruner, J. S., & Ross, G. (1976). The role of tutoring in problem solving. Journal of Child Psychology and Psychiatry, 17(2), 89-100. doi:10.1111/j.1469-7610.1976.tb00381.x
15. Holmes, N. G. (2011). The invention support environment: using metacognitive scaffolding and interactive learning environments to improve learning from invention (Master's dissertation, University of British Columbia).
16. Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K. R. (2009). A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education*, *19*(2), 105-154.
17. Roll, I., Aleven, V., & Koedinger, K. R. (2011). Outcomes and mechanisms of transfer in Invention activities. In *Proceedings of the 33rd annual conference of the cognitive science society* (p. 2824-2829.

Embedded Scaffolding for Reading Comprehension in Open-Ended Narrative-Centered Learning Environments

Jonathan P. Rowe, Eleni V. Lobene, Bradford W. Mott, and James C. Lester

Department of Computer Science, North Carolina State University, Raleigh, NC 27695
{jprowe, eleni.lobene, bwmott, lester}@ncsu.edu

Abstract. Narrative-centered learning environments tightly integrate educational subject matter and interactive stories, where students serve as active participants in story-centric problem-solving scenarios. Embedding scaffolding within the storyline of a narrative-centered learning environment is a discreet approach to supporting students' learning processes without diminishing the motivational benefits of interactive narratives. This paper presents an implementation of story-embedded scaffolding in a narrative-centered learning environment, CRYSTAL ISLAND. CRYSTAL ISLAND's curricular focus has recently been expanded to include literacy education, with a focus on reading. Scaffolding takes the form of *concept matrices*, which are student-generated graphic organizers for complex informational texts that students read as part of CRYSTAL ISLAND's interactive narrative plot. Leveraging generative learning theory, we discuss directions for fading concept matrix-based scaffolding, and examine technical challenges and potential solutions.

Keywords: Narrative-centered learning environments, scaffolding, reading.

1 Introduction

There is growing evidence that narrative-centered learning environments, a class of game-based learning environments that embed educational content in interactive story scenarios, are an effective medium for fostering student learning and engagement [1–2]. A key benefit of narrative-centered learning environments is their capacity to discreetly support students' learning processes by tightly integrating educational and narrative elements. Guiding student problem solving in open-ended narrative-centered learning environments is particularly important, because students often have varying degrees of competency at solving ill-structured problems. Consequently, scaffolding in narrative-centered learning environments should meet at least two requirements: scaffolding should be dynamically tailored to individual students, and scaffolding should be naturalistically embedded within interactive narratives in order to sustain student engagement.

This paper proposes extensions to an open-ended narrative-centered learning environment, CRYSTAL ISLAND, that incorporate story-embedded scaffolding features for literacy education using generative graphic organizers. In CRYSTAL ISLAND, reading comprehension is critical for students gathering clues to solve a science

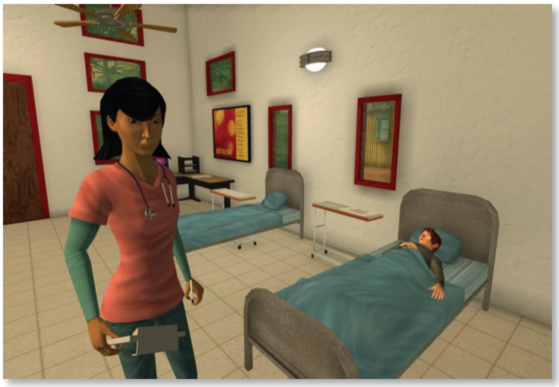


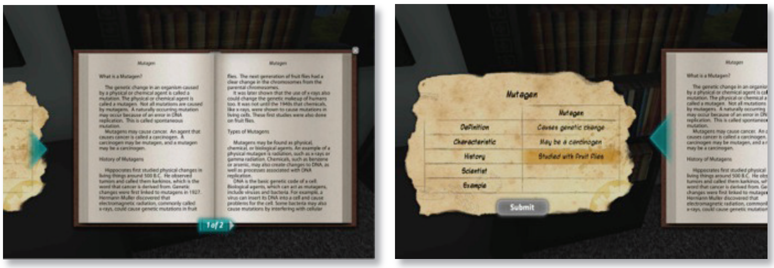
Fig 1. CRYSTAL ISLAND narrative-centered learning environment.

problem-solving mystery. Adaptively scaffolding students' reading processes is a promising direction for enhancing students' literacy skills, and has been the subject of considerable research by the intelligent tutoring systems community [3–4]. We describe how CRYSTAL ISLAND's plot and game mechanics currently incorporate story-embedded graphic organizers to scaffold students' reading comprehension processes, and outline future directions for intelligently diagnosing and fading this scaffolding.

2 CRYSTAL ISLAND for Literacy Education

Over the past several years, our lab has been developing CRYSTAL ISLAND (Fig. 1), a narrative-centered learning environment for middle school microbiology [1]. CRYSTAL ISLAND's curricular focus has recently been expanded to include literacy education based on Common Core State Standards. CRYSTAL ISLAND's narrative focuses on a spreading illness afflicting a research team on a remote island. Students act as medical detectives who must diagnose and treat the illness to save the team.

As part of CRYSTAL ISLAND's curricular focus on literacy, students encounter books and articles throughout the camp that contain complex informational texts about microbiology concepts (Fig. 2, left). Students read and analyze these texts, as well as complete associated concept matrices, to acquire knowledge to diagnose the illness. Concept matrices (Fig. 2, right) are graphic organizers, which students use to record key pieces of information encountered in the informational texts. The concept matrices are framed within the narrative as partially completed notes written by one of the research team's sick scientists. Students must discover and "complete" the notes based on content in the informational texts. The graphic organizers serve both as scaffolds for reading comprehension, as well as embedded assessments of



Mutagen

What is a Mutagen?

The genetic change in an organism caused by a physical or chemical agent is called a mutation. The physical or chemical agent is called a mutagen. Not all mutations are caused by mutagens. A naturally occurring mutation may occur because of an error in DNA replication. This is called spontaneous mutation.

Mutagens may cause cancer. An agent that causes cancer is called a carcinogen. A carcinogen may be mutagen, and a mutagen may be a carcinogen.

History of Mutagens

Hippocrates first studied physical changes in living things around 500 B.C. He observed tumors and called them karkinos, which is the word that cancer is derived from. Genetic changes were first linked to mutagens in 1927. Hermann Muller discovered that electromagnetic radiation, commonly called x-rays, could cause genetic mutations in fruit flies. The next generation of fruit flies had a clear change in the chromosomes from the parental chromosomes.

It was later shown that the use of x-rays also could change the genetic makeup of humans too. It was not until the 1940s that chemicals, like x-rays, were shown to cause mutations in living cells. These first studies were also done on fruit flies.

Types of Mutagens

Mutagens may be found as physical, chemical, or biological agents. An example of a physical mutagen is radiation, such as x-rays or gamma radiation. Chemicals, such as benzene or arsenic, may also create changes to DNA, as well as processes associated with DNA replication.

DNA is the basic genetic code of a cell. Biological agents, which can act as mutagens, include viruses and bacteria. For example, a virus can insert its DNA into a cell and cause problems for the cell. Some bacteria may also cause mutations by interfering with cellular

Mutagen

	Mutagen
Definition	Causes genetic change
Characteristic	May be a carcinogen
History	Studied with Fruit Flies
Scientist	
Example	

Fig 2. (Left) An informational text stylistically formatted like a virtual book, and (Right) a concept matrix stylistically formatted as a scrap of note paper.

students' reading comprehension skills. Completing a concept matrix involves clicking on each blank cell and selecting responses from drop-down menus. After a student has filled out a concept matrix, she can press an on-screen "Submit" button to receive immediate feedback on her responses.

3 Story-Embedded Scaffolding for Reading Comprehension

Graphic organizers, such as concept matrices, provide a natural mechanism for scaffolding reading comprehension skills in a non-obtrusive manner within narrative-centered learning environments. However, generative learning theory suggests that students will achieve improved learning gains if they create the concept matrices themselves. The current implementation of concept matrices in CRYSTAL ISLAND is highly structured. We plan to extend the current approach by intelligently reducing concept matrices' pre-specified structure as students improve their reading skills. Specifically, we propose fading the story-embedded scaffolding by transitioning from highly structured concept matrices to increasingly student-generated concept matrices.

Currently, whenever a student encounters a concept matrix in the story world, the matrix's layout (i.e., number of columns, number of rows) is fixed, the headings are pre-specified, and the set of possible answers for each cell are given. Fading the structure of story-embedded concept matrices can occur in at least three stages. First, one could remove the multiple-choice response menus for interior cells, instead requiring students to enter free-form text. This would require students to independently identify relationships between key terms and concepts from informational texts. Second, one could remove the pre-specified headers for each column and row, replacing them with either multiple-choice menus or free-form text entries. This would require students to independently identify the important themes in informational texts. Third, one could require students to specify the concept matrix layouts by selecting their number of columns and rows. This would require students to independently evaluate which, and how many, themes are most salient.

Effectively fading concept matrix-based scaffolding within CRYSTAL ISLAND raises notable technical challenges. The first challenge is identifying when to transition between successive levels of fading. This could be implemented as a fixed progression (e.g., if the student has encountered *N* concept matrices, fade by one level). Alternatively, fading decisions could be based on probabilistic student models—a common practice in ITSs—although assessing student knowledge from concept matrices presents its own challenges. One could also leverage reinforcement learning to induce optimal fading policies from an exploratory corpus of student interaction data, a technique that has shown success in tutorial dialogue modeling [5].

A second challenge is automatically assessing the quality of student-generated concept matrices. Automated assessment would require models of important concepts and themes from informational texts, as well as robust techniques for comparing informational text models to student-generated concept matrices, which may suffer from spelling errors, misconceptions, and incompleteness. Third, providing feedback tailored to individual students based on their self-generated concept matrices is difficult. Feedback could concern a broad range of subjects, such as corrections of factual errors, clarifications about important themes, or suggestions for alternate layouts, and it would need to cope with students' free-form written content.

Automated assessment and feedback raise interesting computational challenges, but intermediate solutions may exist. For example, it seems plausible that one could identify constraints that good concept matrices meet (e.g., included content terms, content of rows/columns), suggesting that constraint-based models [6] may show promise. While the computational challenges are substantial, tailoring and fading generative graphic organizers to scaffold reading comprehension in open-ended narrative-centered learning environments shows considerable promise for promoting both effective and engaging literacy learning experiences.

Acknowledgments. This research was supported by the National Science Foundation under Grant DRL-0822200. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Additional support was provided by the Bill and Melinda Gates Foundation, the William and Flora Hewlett Foundation, and EDUCAUSE.

References

1. Rowe, J.P., Shores, L.R., Mott, B.W., Lester, J.C.: Integrating Learning, Problem Solving, and Engagement in Narrative-Centered Learning Environments. International Journal of Artificial Intelligence in Education. 21, 115–133 (2011)
2. Johnson, W.L.: Serious Use of a Serious Game for Language Learning. International Journal of Artificial Intelligence in Education. 20, 175–195 (2010)
3. Chen, W., Mostow, J., & Aist, G.S.: Recognizing Young Readers' Spoken Questions. International Journal of Artificial Intelligence in Education. 21 (2011)
4. Dela Rosa, K. & Eskenazi, M.: Self-Assessment of Motivation: Explicit and Implicit Indicators in L2 Vocabulary Learning. In: 15th International Conference on Artificial Intelligence in Education, pp. 296–303. Springer-Verlag, Berlin Heidelberg (2011)
5. Chi, M., VanLehn, K, Litman, D., Jordan, P.: Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical tactics. User Modeling and User Adapted Instruction, 21, 137–180 (2011)
6. Mitrovic, A.: Fifteen years of constraint-based tutors: What we have achieved and where we are going. User Modeling and User-Adapted Interaction. 22, 39–72 (2011)

Suggest-Assert-Modify: A Taxonomy of Adaptive Scaffolds in Computer-Based Learning Environments

James R. Segedy, Kirk M. Loretz, and Gautam Biswas

Institute of Software Integrated Systems, Department of Electrical Engineering and Computer Science, Vanderbilt University, 1025 16th Avenue South, Nashville, TN, 37212, U.S.A.
{james.segedy,kirk.m.loretz,gautam.biswas}@vanderbilt.edu

Abstract. Adaptive scaffolding in computer-based learning environments (CBLEs) continues to be an active area of research, with researchers framing the problem as determining the *what*, *when*, *how*, and *by whom or what* of adaptive scaffolding strategies. This paper presents our recent work in developing a taxonomy for adaptive scaffolds in CBLEs. The taxonomy, motivated by previous work in developing adaptive scaffolds, attempts to address the *how* of scaffolding by describing the tools and techniques available for scaffolding in CBLEs. We present the taxonomy, which describes adaptive scaffolds as one or more suggestions, assertions, and learning task modifications, and we discuss the utility of the taxonomy in describing adaptive scaffolding strategies.

Keywords: adaptive scaffolds, taxonomy, computer-based learning environments

1 Introduction

Research in computer-based learning environments (CBLEs) has long recognized the vital role of adaptivity in the success of a system's ability to independently foster learning in students [1]. Adaptive CBLEs regularly capture and analyze student activities in order to make decisions about how and when to scaffold learners [2]. These systems *take explicit actions* [3]; they may remind learners of relevant information, advise learners on how to proceed in their learning tasks, or modify the difficulty level of the learning activity itself.

The methods and tools used for scaffolding may vary widely based on the goal of instruction. For example, Chi and colleagues [4] presented 15 types of scaffolding actions identified in the research literature. These scaffolds include providing hints, fill-in-the-blank prompts, explanations, and correct answers, among others. Understanding these techniques, including when and why a particular scaffold may be more effective than another, remains an important area of research. Pea [5] framed the problem as defining the *what*, *why*, and *how* of scaffolding. *What* information should a scaffolding action focus on, w*hy* should a CBLE employ a scaffold, and *how* does the CBLE actually scaffold the learner (*i.e.*, what action does it take)? This framework was later revised by Azevedo & Jacobson [2] to focus on *what*, *when*, *how*, and *by whom or what*. The revised framework replaces the *why* question with a *when* question: *when* should a CBLE scaffold learners? It also introduces a new question: who or what should provide the scaffolds?

In this paper, we attempt to address the *how* question by presenting a novel taxonomy for classifying adaptive scaffolds in CBLEs. The taxonomy classifies adaptive scaffolds as a set of one or more *suggestions*, *assertions*, and learning task *modifications* (SAMs). Section 2 presents the background and motivation for the taxonomy; section 3 presents the taxonomy; and section 4 discusses future directions.

2 Previous Work in Classifying Adaptive Scaffolds

While some researchers in the field of educational technology have proposed methods for classifying and describing adaptive scaffolding approaches based on well-defined terms (*e.g.*, [6-7]), no comprehensive taxonomy of the tools and techniques available for scaffolding currently exists. Thus, the field now suffers from a lack of operational definitions, and several researchers refer to the scaffolds in their systems as "hints" or "feedback." Often, researchers define these scaffolds via examples. Bell & Davis [8], for instance, differentiate between three types of hints provided by a pedagogical agent named Mildred: activity hints, evidence hints, and claim hints. The provided descriptions of the hints are vague, and they are mainly illustrated with examples:

> The current instantiation of Mildred provides three types of hints - on activities, evidence, and claims. For example, in the "Critique Evidence" activity of All The News, an activity hint might say, "When you critique the evidence, you will think about: (1) the science ideas used in the evidence, (2) the methods used to create the evidence, and (3) how credible or believable the evidence is." Further activity hints for the Critique Evidence activity would provide definitions and examples of the critique criteria of science, methods, and credibility. Evidence hints are more specific, providing help in thinking about a particular piece of evidence. A hint for the "Bicyclists at Night" evidence (used in both All The News and How Far) is, "Why is the person in white [clothes] easier to see? What is happening to the light?" A student working on a critique of the Bicyclists at Night evidence could then receive converging evidence on both the act of critiquing and the specific evidence being critiqued. Likewise, claim hints help students think about a particular claim. For example, a claim hint about black "attracting heat" (as opposed to absorbing light) might say, "What would happen if there were a heat source in a dark room? Would someone wearing black get hotter than someone wearing white?" (p. 144)

Similarly, Jackson, Guess, & McNamara [9] present a CBLE, *iStart*, and describe the scaffolds provided by the system as "feedback" without defining the term, instead relying on examples:

> Merlin provides feedback for each explanation generated by the student. For example, he may prompt them to expand the explanation, ask the students to incorporate more information, or suggest that they link the explanation back to other parts of the text. Merlin sometimes takes the practice one step further and has students identify which strategies they used and where they were used. (p. 129)

Some researchers have developed more specific scaffold classifications. For example, Belland, Glazewski, & Richardson [10] propose four types of scaffolds: conceptual support, metacognitive support, procedural support, and strategic support. These support types are defined as help about "what to consider," "how to manage the learning process," "how to use tools," and "what strategies to use in approaching the problem," respectively. This classification differentiates scaffolds based on a single dimension: the type of information the scaffold is designed to support. However, because scaffolds are *actions*, an appropriate classification needs to consider both what information is supported and how it is supported.

In presenting a general framework for the design of Intelligent Tutoring Systems (ITSs), VanLehn [6] defines minimal feedback and three types of hints: point, teach, and bottom out. In ITSs, learners are presented with small multi-step problems in a well-defined domain (*e.g.*, physics). When students are having trouble correctly completing a problem step, the system usually intervenes to provide one of these types of scaffolds. Minimal feedback scaffolds indicate whether or not a learner's attempt at completing a problem step is correct or incorrect. Hints are provided in relation to a particular knowledge component (*e.g.*, a fact, definition, or procedure), and they are defined as follows:

> Pointing hints mention problem conditions that should remind the student of the knowledge component's relevance. Teaching hints describe the knowledge component briefly and show how to apply it. Bottom-out hints tell the student [how to apply the knowledge component to solve] the [current problem] step. (p. 242)

This scaffold classification, unlike the classification described in [10], does focus both on the information the scaffold is designed to support and the methods by which the information is supported. However, it is not general enough to classify a number of scaffolds that have been implemented in CBLEs. For example, several CBLEs provide scaffolds that suggest the use of a particular resource within the system rather than mentioning or explaining a knowledge component.

As a final example, Graesser & McNamara [7] describe the scaffolds implemented within a CBLE called *AutoTutor*, which teaches physics by posing questions and then holding natural language dialogues with learners as they attempt to answer those questions. During the course of these dialogues, *AutoTutor* may employ any of five types of dialogue moves: pumps, hints, prompts, correctness feedback, and assertions. *Pumps* ask the learner to continue elaborating on the answer they have started to offer. For example, *AutoTutor* might encourage a student to "keep going." *Hints* are questions that attempt to elicit a question-relevant proposition from the learner. For example, *AutoTutor* may ask students how Newton's second law of motion applies to the current question. *Prompts* are questions that ask the learner to provide explicit words or phrases that are important in answering the current question. For example, *AutoTutor* may present a partial definition of Newton's second law of motion and ask the learner to fill in the missing information. *Feedback* indicates whether the learner's answer is correct or incorrect, and *assertions* communicate entire propositions to learners when hints and prompts fail to elicit them.

In considering the presented scaffold classifications, some common themes emerge. First, several of the presented scaffolds operate by *providing a suggestion*. For example, pointing hints in ITSs direct attention to specific problem features, suggesting that learners consider those features; Merlin suggests that learners link their current explanation back to other parts of the text; and *AutoTutor* pumps learners, suggesting that they continue elaborating on their answer. Second, several of the presented scaffolds operate by *asserting information*. For example, teaching hints assert knowledge components and how to apply them; bottom-out hints assert how to solve the current problem step; and *AutoTutor's* assertions communicate question-relevant propositions to learners. Third, some scaffolds operate by *modifying the learning task*. For example, when *AutoTutor* asks the learner a question as part of delivering a hint, it is redirecting the learner's attention away from their former task (answering the original question) to a new task (answering a related question).

These observations have led us to develop a taxonomy that classifies adaptive scaffolds as one or more suggestions, assertions, and learning task modifications. This taxonomy is general and widely-applicable. Moreover, it provides a language for presenting and communicating scaffolding strategies.

3 The Suggest-Assert-Modify Taxonomy

The Suggest-Assert-Modify (SAM) taxonomy is illustrated in Figure 1. Suggestion scaffolds provide information to learners for the purpose of prompting them to engage in a specific behavior (*e.g.*, accessing a resource). By executing the recommended behavior, learners should encounter critical information that, if properly internalized, would allow them to make progress in accomplishing the learning task. The taxonomy classifies suggestions based on whether they target metacognitive activities (*e.g.*, planning or reflection) or cognitive knowledge integration activities. Knowledge integration is the process of analyzing and connecting multiple chunks of information in order to achieve new understandings about how they are related [11-12]. It can target several cognitive processes, such as: (i) goal orientation, in which learners integrate chunks of information with their understanding of their current goal; (ii) explanation construction, in which learners assemble chunks of information to explain a system, process, or phenomenon; (iii) prediction, in which learners integrate chunks of information with a hypothetical scenario, and several others.

Assertion scaffolds communicate information to learners as being true; ideally, learners will integrate this information with their current understanding as they continue working toward completing their learning task. Unlike suggestions, assertion scaffolds don't directly encourage learners to engage in a particular behavior; they only state information.

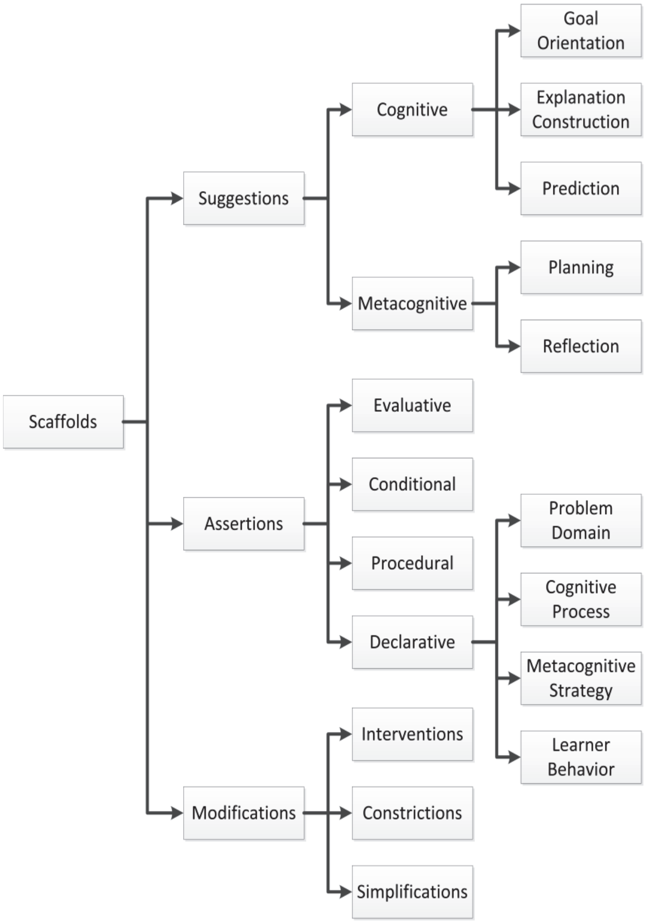


Fig. 1. The SAM Taxonomy for Adaptive Scaffolds

The taxonomy distinguishes between four types of assertion scaffolds: declarative, procedural, conditional, and evaluative. Declarative assertions communicate "knowing that" information [11]. Such information is often conceptualized as being represented as and with schemata: mental structures that represent a concept and the features that characterize it [12]. For example, a schema representing an animal might contain features such as the animal's number of legs and the sound that the animal makes. Features correspond to variables in an algebra expression or computer program; they can take on any of a number of values when instantiated; and an "instance" of an animal schema may represent an actual animal in the world. Thus, declarative assertions contain information that may be represented by a schema; this includes facts, definitions, concepts, and understandings of relationships and interrelationships among actors in complex systems. In the proposed taxonomy, declarative assertions are sub-divided based on their topic, which may be the problem domain, cognitive processes, metacognitive strategies, and the learner's behavior while using the system. Examples of each type of declarative assertion are listed in Table 1.

Assertion Category	Example
Declarative – Problem Domain	Sunfish eat mosquito fish.
Declarative – Cognitive Processes	You have to know how to multiply fractions.
Declarative – Metacognitive Strategies	The "cross-multiply" strategy may help you.
Declarative – Learner Behavior	You haven't tried any division problems.
Procedural	To multiply fractions, first multiply the numerators, and then multiply the denominators.
Conditional	The "cross-multiply" strategy should be used whenever you need to solve for an unknown value in an equation consisting of only fractions.
Evaluative	You don't seem to have a good understanding of how to divide fractions.

Table 1. Types of Assertion Scaffolds with Examples.

Procedural assertions communicate "how-to" information: sets of actions that, when executed in a loosely-ordered sequence, can accomplish a task. These assertions explain how to perform cognitive processes, such as identifying important information in text passages or applying causal reasoning to answer hypothetical questions. Conditional assertions communicate information represented as "if-then" rules that identify both when cognitive processes are applicable and whether or not they should be executed based on the current context [12]. These assertions usually explain metacognitive strategies. In a fractions learning environment, for example, the system might assert that a good strategy for solving algebraic expressions that consist entirely of fractions is to use a "cross-multiply" strategy. This would be represented as the following "if-then" rule: *IF you want to solve an algebraic expression consisting entirely of fractions, THEN employ the cross-multiply strategy*. Finally, evaluative assertions communicate evaluations of the learner's performance and understanding. For example, the system may assert that the learner does not seem to understand how to divide fractions.

Modification scaffolds, unlike suggestion and assertion scaffolds, do not operate by communicating information to the learner; rather, they change aspects of the learning task itself. In doing so, they seek to adapt the task to the learner's needs and abilities. The taxonomy differentiates between three types of modification scaffolds: simplifications, constrictions, and interventions. Simplification modifications, as specified by Wood, Bruner, & Ross [13], operate by "reducing the number of constituent acts required to reach solution." Constriction modifications operate by reducing the number of options available to the learner. For example, the scaffolding agent may block access to tools or resources in order to focus learners' attention on other, more useful approaches to solving the task. Intervention scaffolds, rather than modifying features of the overall task, operate by temporarily shifting learners' attention from their primary task to an intervention task. Upon completion of the intervention task, learners may return to the primary task.

The SAM taxonomy addresses the *how* of scaffolding by describing the atomic elements of adaptive scaffolds, and it provides a language for communicating both individual scaffolds and entire scaffolding strategies. For example, the scaffolding strategy for ITSs discussed by VanLehn [6] could be described as a progression from cognitive suggestions (pointing hints) to declarative assertions that describe a knowledge component (teaching hints) to declarative assertions that provide the answer to the current problem step (bottom-out hints). In comparison to the scaffolding classifications presented in Section 2, we argue that the SAM taxonomy is more comprehensive and general than its predecessors.

4 Conclusion

This paper has presented a novel taxonomy for describing and classifying adaptive scaffolds in computer-based learning environments. The taxonomy classifies adaptive scaffolds as one or more suggestions, assertions, and learning task modifications, and it provides a general, widely-applicable language for communicating and interpreting scaffolding strategies.

The SAM taxonomy, however, is not without limitations. First, the distinction between suggestions and assertions is sometimes ambiguous, and a scaffold may consist of an assertion that implies a suggestion. For example, a scaffold in an algebra learning environment may assert that successful students used a particular problem solving strategy in order to indirectly suggest that the learner adopt that strategy. Second, the SAM taxonomy does not currently distinguish between different types of intervention scaffolds. Future work should investigate methods for breaking down interventions according to the types of activities learners are expected to accomplish during the intervention. For example, it may be valuable to separate modeling interventions (*e.g.*, demonstrating how to solve a problem), metacognitive interventions (*e.g.*, requiring learners to gauge their own comprehension), and cognitive interventions (*e.g.*, requiring learners to correctly define terms or explain properties of a complex system).

It is important to note that the presented taxonomy represents an initial step toward a standardized language for describing the *how* of adaptive scaffolding strategies. As we continue to scan the literature for more examples of adaptive scaffolds in educational technology, we will update the taxonomy as needed to reflect distinguishing features of adaptive scaffolds.

Acknowledgements. This work has been supported by Institute of Educational Sciences CASL Grant #R305A120186

References

1. Park, O.C., Lee, J.: Adaptive Instructional Systems. In: Jonassen, D.H. (ed.) Handbook of Research for Education Communications and Technology, pp. 651-684. Erlbaum, Mahwah (2004)
2. Azevedo, R., Jacobson, M.J.: Advances in scaffolding learning with hypertext and hypermedia: A summary and critical analysis. Educational Technology Research and Development, 56, 93-100 (2008)
3. Puntambekar, S., Hübscher, R.: Tools for scaffolding students in a complex learning environment: What have we gained and what have we missed? Educational Psychologist, 40, 1-12 (2005)
4. Chi, M.T.H., Siler, S.A., Jeong, H., Yamauchi, T., Hausmann, R.G.: Learning from human tutoring. Cognitive Science, 25, 471-533 (2001)
5. Pea, R.D.: The social and technological dimensions of scaffolding and related theoretical concepts for learning, education, and human activity. The Journal of the Learning Sciences, 13, 423-451 (2004)
6. VanLehn, K.: The behavior of tutoring systems. International Journal of Artificial Intelligence in Education, 16, 227–265 (2006)
7. Graesser, A.C., McNamara, D.: Self-regulated learning in learning environments with pedagogical agents that interact in natural language. Educational Psychologist, 45, 234-244 (2010)
8. Bell, P., Davis, E.A.: Designing Mildred: Scaffolding students' reflection and argumentation using a cognitive software guide. In: Fishman, B., O'Connor-Divelbliss, S. (eds.) Fourth International Conference of the Learning Sciences, pp. 142-149. Erlbaum, Mahwah (2000)
9. Jackson, G.T., Guess, R.H., McNamara, D.S.: Assessing Cognitively Complex Strategy Use in an Untrained Domain. Topics in Cognitive Science, 2, 127-137 (2010)
10. Belland, B.R., Glazewski, K.D., Richardson, J.C.: A scaffolding framework to support the construction of evidence-based arguments among middle school students. Educational Technology Research and Development, 56, 401-422 (2008)
11. Anderson, J.R.: ACT: A simple theory of complex cognition. American Psychologist, 51, 355-365 (1996)
12. Winne, P.H.: Self-regulated learning viewed from models of information processing. In: Zimmerman, B.J., Schunk, D.H. (eds.) Self-Regulated Learning and Academic Achievement: Theoretical Perspectives, pp. 153-189. Erlbaum, Mahwah (2001)
13. Wood, D., Bruner, J.S., Ross, G.: The role of tutoring in problem solving. Journal of Child Psychology and Psychiatry, 17, 89-100 (1976)

Exploring Adaptive Scaffolding in a Multifaceted Tangible Learning Environment

Elissa Thomas, Victor Girotto, Alex Abreu, Cecil Lozano, Kasia Muldner, Winslow Burleson, Erin Walker

Computing, Informatics & Decision Systems Engineering, Arizona State University

{eethomas, victor.girotto, alexabreu, cecil.lozano, katarzyna.muldner, winslow.burleson, erin.a.walker}@asu.edu

Abstract. The majority of educational software is designed for traditional computers, which allow little opportunity for physical manipulation of an environment. Tangible Activities for Geometry (TAG) provides students a tangible learning environment. Currently, however, TAG does not employ adaptive scaffolding techniques. Accordingly, we describe how scaffolding techniques and teachable agent behaviors can be integrated into TAG to improve this tangible learning environment.

Keywords: adaptive scaffolding, tangible learning environments, teachable agents

1 Introduction

Open-ended learning environments (OELEs) enable students to actively engage in problem solving, such as generation, testing and revision of a hypothesis [1]. However, most educational systems target personal computers and their typical WIMP (window, icon, menu, pointing device) setup. These systems rarely allow for embodied interaction between the student and the learning environment, despite the fact that students learn a great deal through physically engaging with their environment [2]. The *Tangible Activities for Geometry* system (TAG) aims to fill this gap, by providing a tangible OELE where students can move beyond the boundaries of the virtual world and explore different strategies for solving geometric problems [3].

The current TAG system provides no feedback or adaptation to the user's performance. Therefore, our goal with this paper is to propose ways of integrating adaptive scaffolding techniques into this tangible learning environment (TUI), laying the foundation for studying the effects that they would have in this type of learning environment. The majority of TUIs do not currently possess such capabilities, which allows us to start exploring this intersection. Here, we will review existing frameworks and techniques that can be used for scaffolding the user's learning in an adaptive manner and will describe ways in which they could be applied to our system.

2 Description of Current System

In the current implementation of the TAG system, a student solves geometry problems by instructing a teachable agent on the steps needed to solve the problem. Problems include plotting a point in a given quadrant, translating a point, or rotating a point around a center of origin. While answers are sometimes the same, problems can often be solved in different ways. The system is comprised of three main components [3]. The *problem space* is a Cartesian plane projected on the ground. This is where the teachable agent and the problem objects, such as lines and points, are displayed. The interactions with the problem space occur through a *hanging pointer* that hangs from the ceiling, functioning as a mouse. Hovering the pointer over the problem space moves the cursor. Clicking is performed when the user moves the pointer below a certain height threshold and back up. The feedback for the user's interactions on the problem space is received on the *mobile interface,* displayed on an iPod Touch. In this interface, the user is able to select an action that will be performed by the agent, view the steps already taken, and navigate through problems.

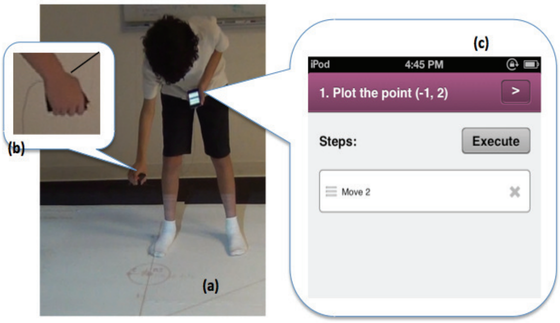


Figure 1: Elements of the TAG system. The problem space (a), where the Cartesian plane is projected, the hanging pointer (b), used by the student to interact with the problem and the mobile interface (c), the iPod interface commands are issued to the agent.

3 Review of Existing Pedagogical Techniques

Prior research has explored how various pedagogical techniques impact student learning. A number of these rely on a teachable agent paradigm, where students learn by tutoring a computerized agent modeled to simulate behaviors of a student tutee. For instance, reflective knowledge building uses questions and explanations generated by a teachable agent to prompt students to reflect on their own understanding of various concepts, and refine their ideas [4]. Agents could also use this technique to introduce new ideas to a student's existing knowledge [5].

Other research has shown that the level of abstraction in the advice provided by a teachable agent can impact a student's perceptions and performance. Students who work with agents that give different kinds of feedback, ranging from high-level advice to concrete, task specific suggestions, performed better than students who interacted with agents that only used task-specific suggestions [6].

Techniques used in cognitive tutors can also be useful for extending TAG. Cognitive tutors provide the user with feedback on a step-by-step basis, in response to common errors and with on-demand instructional hints, and adapt the selection of problems based on user-performance [7]. The challenge is to adapt these techniques to an open-ended system such as TAG while still encouraging open-ended exploration.

4 Proposed Extensions on the Current System

We propose expanding TAG to employ adaptive scaffolding as a way to increase the system's effectiveness. Techniques such as reflective knowledge building could be integrated into our system to improve student learning while also enhancing unique tangible aspects of our system. For example, if the student is attempting to plot a point in quadrant II, but moved the agent into quadrant IV, a question from the agent might prompt the student to recognize that their actions are not leading them to the correct solution. As another example, after a student solves a problem, the TAG agent could propose an alternate solution, helping students evolve their ideas, which some students struggle to do in OELEs [8]. As an extension of adaptive scaffolding in a traditional learning environment, students could also be encouraged to try additional tangible interactions that may not have been incorporated into their original solution.

Scaffolding could also be employed through hints given by the agent while a student is working on a problem. In this scenario, the agent uses cues that a student might be confused, such as a long pause without any activity, and provides a hint to guide the student in the right direction. Are there unique cues within TUIs that could be detected to improve an adaptive scaffolding model? To study this, our system could monitor embodied behaviors exhibited by the student, such as pacing back and forth or kneeling down on the projected Cartesian plane. Following standard convention, the agent's hints should vary in detail based on the student's performance within a given problem. Students would initially be provided with high-level feedback from the teachable agent, allowing them to apply the information given to them by the agent to the problem domain. If the student continues having trouble, the system can adaptively adjust the agent's hints to be more direct, allowing students to discover the correct approach, albeit, with less reflection on the metacognitive process. By providing feedback in this manner, we can foster an atmosphere of discovery, which should help students feel more engaged [2]. Since previous work has shown that increasing the sociability of an agent improves student perceptions of the system and student performance [9], hints from the agent could be provided textually through a pop up on the iPod interface while also being spoken by the agent.

On a less localized scale, adaptive scaffolding could also be applied based on a student's performance throughout an entire session. Indicators that could be used to measure student performance include the amount of time taken to solve a problem, the number of correct and incorrect solutions a student has produced, and the number of steps a student uses as compared to an optimal solution with a minimal number of steps. Applying this type of adaptive scaffolding in a TUI introduces some unique challenges. For example, how do we differentiate between students that are struggling with the problem domain and students that are having trouble understanding how to use the unique tangible interactions of our system?

5 Conclusion

By proposing a novel set of techniques to augment the TAG system, we aim to provide the appropriate level of scaffolding needed to improve student learning, while maintaining student engagement when faced with difficulties and failure. The ultimate goal is to ensure that students receive help when it is needed, but are not hindered during open-ended exploration. We also hope to learn more about how this scaffolding should be presented to the student on the different dimensions that a TUI provides, exploring the advantages and drawbacks of each type of scaffolding.

References

1. Land S. Cognitive requirements for learning with open-ended learning environments. Educational Technology Research and Development. 2000, Volume 48, Issue 3, pp 61-78.
2. Walker, E, and Burleson, W. User-Centered design of a teachable robot. Intelligent Tutoring Systems, 2012.
3. Mulder, K., Lozano, C., Girotto, V., Burleson, W., and Walker, E. Designing a Tangible Learning Environment with a Teachable Agent. Artificial Intelligence in Education, 2013.
4. Roscoe, D., Wagster, J., and Biswas, G., Using Teachable Agent Feedback to Support Effective Learning by Teaching, In Proceedings of the 30th Annual Meeting of the Cognitive Science Society, Washington, DC, 2008.
5. Blair, K., Schwartz, D., Biswas, G., and Leelawong, K. Pedagogical Agents for Learning by Teaching: Teachable Agents. In Educational Technology & Society: Special Issue on Pedagogical Agents, 2006.
6. Lester, J. C., Converse, S. A., Kahler, S. E., Barlow, S. T., Stone, B. A., and Bhogal, R. S. The Persona Effect: Affective Impact of Animated Pedagogical Agents. In Proceedings of CHI ‘97, 1997.
7. Koedinger, K., Aleven, V. Exploring the Assistance Dilemma in Experiments with Cognitive Tutors. 2007.
8. Land, S. M. Cognitive Requirements for Learning with Open-Ended Learning Environments. Educational Technology Research and Development 48.3, 2000.
9. Hershey D. K., Mishra P., and Altermatt, E. All or nothing: Levels of sociability of a pedagogical software agent and its impact on student perceptions and learning. Journal of Educational Multimedia and Hypermedia 14.2, 2005.

“Gaming the system” in Newton’s Playground

Lubin Wang, Yoon Jeon Kim, & Valerie Shute

Florida State University, Tallahassee, USA
lw10e@fsu.edu,yk06c@my.fsu.edu,vshute@fsu.edu

Abstract. This paper describes the current status of ongoing research looking into students’ “gaming the system” behaviors in an open-ended learning environment—the game Newton’s Playground—in relation to their physics learning, enjoyment of the game, and persistence. Our next step is to code students’ gaming behaviors and then compare learning via pretest and posttest scores. We’ll also examine gaming behaviors relative to enjoyment of the game and persistence. Findings can inform improvements to Newton’s Playground (and other games) and guide the design of scaffolding for students in other OELEs.

Keywords: game the system behaviors, game-based learning, physics learning, persistence

1 Introduction

Open-ended learning environments (OELEs) are technology-rich environments that allow learners to participate in authentic problem solving activities, interact with the system by actively making choices, and apply cognitive and metacognitive skills to assess and monitor their learning processes [5]. Providing players the freedom to explore the environment and make choices are essential features of OELEs, which render the environment engaging and meaningful.

Well-designed digital games share similar features with such environments [1]. For example, Gee (2003) discusses properties of good games, such as interactive problem solving, adaptive challenges, feedback, and control that are aligned with learning principles to promote deep and meaningful learning. In games players actively interact with the system by making choices, and this provides a sense of control and ownership to the players. Also, games provide players with complex and interesting problems to solve, allowing freedom in terms of how they reach the solution.

In such wide-open environments, however, it is almost impossible to predict every possible way that learners will interact with the system. Studies have shown that for novice learners, having too much freedom can lead to frustration or unsuccessful learning [5]. This may result in unexpected behaviors by learners such as exploiting loopholes of the system, which is commonly referred to as gaming the system.

Baker (2005) defines gaming the system as “attempting to succeed in an educational environment by exploiting properties of the system rather than by learning the material and trying to use that knowledge to answer correctly (p. 6).” Reasons why learners game the system and how it influences learning have been investigated in various forms of technology rich learning environments, primarily in intelligent tutoring systems [1]. Broadly speaking, learners are more likely to show gaming the system behaviors when (a) they dislike the subject matter, (b) they are frustrated, and/or (c) they lack drive or motivation.

Unlike what happens in learning environments like intelligent tutoring systems, gaming the system is not always viewed negatively in the gaming context. In fact, it can be an important aspect of gaming culture as evidenced by a player proudly sharing certain "tricks" with other players [4]. Therefore, as using games for learning purposes becomes a more common practice in the broader education community, it is important for educators and researchers to understand why players would game the system and how such behavior influences learning.

2 Context

We propose to investigate gaming the system behaviors in a game called Newton's Playground (NP) [6]. NP is a two-dimensional computer game designed to assess and support qualitative physics and persistence. The core mechanic of the game is to guide a green ball to a red balloon by drawing physical objects and simple mechanical devices (i.e., ramp, lever, pendulum, springboard) on the screen that "come to life" once drawn. We call these devices "agents of force and motion" since they trigger or change the direction of motion. There are four types of agents that are categorized in terms of unique features and underlying physics principles: ramp, lever, pendulum, and springboard.

A ramp is any line drawn that guides a ball in linear motion, and it is commonly used for problems that require transfer of potential energy to kinetic energy. A lever rotates around a fixed point usually called a fulcrum or pivot point, and it is used to move the ball vertically. A swinging pendulum directs an impulse tangent to its direction of motion, which is used to exert a horizontal force. A springboard stores elastic potential energy provided by a falling weight, and is used to move the ball vertically.

As the use of these agents provides evidence for students' physics understanding, NP has a built-in evidence identification system that automatically categorizes (with > 95% accuracy when compared with human ratings) the type of agent based on salient features of drawn objects by students. Even though there is no absolute correct or incorrect way of solving problems, they are "probable agents" of force and motion that experts (or the game designers) expect players to use in given problems.

In the fall of 2012, we had 165 ninth graders play the game for around 4 hours (across a one-week time frame). We also administered pre- and posttests of physics to measure improvement of students' qualitative physics as the result of playing NP. As part of the study, we observed that some players came up with various ways to exploit the system, and we categorize them as stacking lines, breaking the system, and cutting corners (Table 1). We define these types of solutions as gaming the system behaviors in NP because these solutions (a) exploit loopholes in the system, and (b) do not require application of appropriate physics principles.

Table 1. Gaming the System in Newton's Playground

Gaming the system behaviors	Features
Stacking	Players consecutively draw short lines right below the ball to lift up the ball to the balloon. Players are likely to show this behavior when the balloon is above the ball.
Breaking the system	Players draw random lines across the given objects until the system crashes and acts randomly. Players are likely to show this behavior when either the balloon is above the ball or the path to the balloon is constrained by obstacles
Cutting corners	Players draw a line quickly beneath the ball that spans over to the balloon. Players are likely to show this behavior when the ball is moving away from the balloon or the starting point of the ball is higher than the balloon.

3 Research Questions

The present study aims to address the following questions:

1. How does gaming the system in NP influence players' physics learning?
2. How does gaming the system in NP relate to players' enjoyment of the game and persistence?

Our hypotheses are:

1. For most students, gaming the system is negatively related to players' physics learning;
2. For most students, gaming the system is negatively related to players' enjoyment of the game and persistence.

4 Method

First, two human raters will replay (with the "level replay" function in the game) all log files of a set of 16 problems that are solved by over 60% of the students, and manually code occurrences of gaming the system behavior related to the three identified categories (i.e., stacking, breaking the system, and cutting corners). Second, we will identify three different subgroups of players in terms of frequencies of the gaming the system behaviors (i.e., none, some, and a lot). Third, we will analyze differences among these subgroups in terms of physics learning (via pretest to posttest gains), enjoyment, and persistence. Note that we already have the data collected, and just need to conduct the observation of replay files, code the behaviors, and analyze the data.

5 Discussion and Implications

To ensure that learners with varying abilities can all benefit from playing games that are designed for learning, we need to identify any subgroups of students who may become lost in the environment and simply try to "cheat through" the problems without applying appropriate knowledge and skills. If our hypotheses are established, we will need to devise appropriate scaffolds in NP to minimize the gaming behavior and thus maximize learning and enjoyment. Potential scaffolds that may fit in NP include tutorial videos and visual aid function. For example, for the visual aid function, dotted lines will show up on the screen upon request, which provide students with clues for appropriate agents rather than having them get stuck and thus frustrated.

However, considering NP is still a game, any decisions regarding scaffolds need to balance with features of good games. That is, we need to be careful about how much scaffolds we provide, and how they are presented to students because poorly designed scaffolds in the game may spoil engaging features of the game (e.g., challenge, control, and adaptive difficulty).

In conclusion, gaming the system behaviors have not been fully investigated in the context of games for learning, and we first need to understand how these behaviors influence learning—i.e., are they always maladaptive or can they sometimes yield positive outcomes? We hope that this study will provide us with useful information about learners' gaming the system behaviors in NP in relation to learning and enjoyment, and also shed light on appropriate forms of scaffolding to be used to prevent such behaviors, if warranted. The findings from this study may also be of interest to researchers who are interested in gaming behaviors and possible scaffolding in OELEs.

References

1. Baker, R. S., Corbett, A. T., Roll, I., & Koedinger, K. R. (2008). Developing a generalizable detector of when students game the system. *User Modeling and User-Adapted Interaction, 18* (3), 287-314.
2. Baker, (2005). *Designing intelligent tutors that adapt to when students game the system (*doctoral dissertation). Carnegie Mellon University, Pittsburgh, PA.
3. Gee, J. P. (2003). *What Video Games Have to Teach Us About Learning and Literacy*? New York: Palgrave Macmillan.
4. Kuecklich, J. (2004). *Other playings— cheating in computer games*. Paper presented at Other Players Conference. Copenhagen, Denmark.
5. Land, S. M. (2000). Cognitive requirements for learning with open-ended learning environments. *Educational Technology Research and Development, 48* (3), 61—78.
6. Shute, V. J., & Ventura, M. (2013). Measuring and supporting learning in games: Stealth assessment. Cambridge, MA: The MIT Press.

AIED 2013 Workshops Proceedings
Volume 3

2nd Workshop on

Intelligent Support for Learning in Groups

Tuesday, July 9, 2013
Memphis TN

Rohit Kumar
Speech, Language and Multimedia
Raytheon BBN Technologies
USA

Jihie Kim
Information Sciences Institute
University of Southern California
USA

https://sites.google.com/site/islg2013/

Preface

Technological advances in the use of Artificial Intelligence for educational applications over the past two decades have enabled the development of highly effective, deployable learning technologies that support learners across a wide-range of domains and age-groups. Alongside, mass access and adoption of revolutionary communication technologies have made it possible to bridge learners and educators across spatio-temporal divides. On the other hand, research in collaborative learning has informed instructional principles that leverage the pedagogical benefits of learning in groups. Educational service providers including mainstream universities are deploying their courses to online learning platforms that allow students to share their learning experience with their peers. Large volumes of educational content including videos, presentations, books and games are accessible on mobile/tablet devices which enrich learning interactions by bringing students together.

Over the past few years, the AIEd research community has started investigating extension of fundamental techniques (such as student modeling, model-based tutors, integrated assessment, tutorial dialog, automated scaffolding, data mining, pedagogical agents) to support learning in groups. The goal of this series of workshops is to provide a focused forum for bringing this sub-community of AIEd researchers together to share recent advances in the field.

Building on its first instantiation in 2012, this workshop will comprise of presentations describing advances in state of the art AIEd techniques to improve the effectiveness of learning in groups. Five full length papers and six short papers were accepted for presentation this year. These eleven papers are organized into four interrelated areas that cover the breadth of the topics of interest. Additionally, two positions papers accepted to this workshop are included in these proceedings. Besides the paper presentations, the workshop will include a group discussion session. After the workshop, notes from this session, will be shared on the workshop website.

June, 2013
Rohit Kumar, Jihie Kim

Organizers

Jihie Kim	*University of Southern California, USA* (jihie@isi.edu)
Rohit Kumar	*Raytheon BBN Technologies, USA* (rkumar@bbn.com)

Advisors

Arthur C. Graesser	*University of Memphis, USA*
Lewis W. Johnson	*Alelo Inc., USA*
James C. Lester	*North Carolina State University, USA*
Carolyn P. Rosé	*Carnegie Mellon University, USA*
Beverly P. Woolf	*University of Massachusetts Amherst, USA*

Program Committee

Ari Bader-Natal	*Minerva Project, USA*
Ryan S.J.d Baker	*Columbia University, USA*
Kristy Boyer	*North Carolina State University, USA*
Stavros N. Demetriadis	*Aristotle University of Thessaloniki, Greece*
Toby Dragon	*Saarland University, Germany*
Gahgene Gweon	*Korea Advanced Institute of Science and Technology, S. Korea*
Sharon I-Han Hsiao	*Columbia University, USA*
Charalampos Karagiannidis	*University of Thessaly, Greece*
Judy Kay	*University of Sydney, Australia*
Fazel Keshtkar	*University of Memphis, USA*
Diane Litman	*University of Pittsburgh, USA*
Bruce M. McLaren	*Carnegie Mellon University, USA* *Saarland University, Germany*
Elaine Raybourn	*Sandia National Labs, USA*
Oliver Scheuer	*Saarland University, Germany*
Erin Walker	*Arizona State University, USA*

Additional Reviewers

Tiffany Barnes	*University of North Carolina at Charlotte, USA*

Table of Contents

Tools & Techniques

Modeling & Analysis

Visualization

Interactive Systems

Position Papers

Authoring Collaborative Intelligent Tutoring Systems

Jennifer K. Olsen[1], Daniel M. Belenky[1], Vincent Aleven[1], Nikol Rummel[12], Jonathan Sewall[1], and Michael Ringenberg[1],

[1]Human Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA, USA
jkolsen@cs.cmu.edu, dbelenky@andrew.cmu.edu,
{aleven,sewall,mringenb}@cs.cmu.edu
[2]Institute of Educational Research, Ruhr-Universität Bochum, Germany
nikol.rummel@rub.de

Abstract. Authoring tools for Intelligent Tutoring System (ITS) have been shown to decrease the amount of time that it takes to develop an ITS. However, most of these tools currently do not extend to collaborative ITSs. In this paper, we illustrate an extension to the Cognitive Tutor Authoring Tools (CTAT) to allow for development of collaborative ITSs that can support a range of collaboration scripts. Authoring tools for collaborative ITSs must be flexible enough to allow for different learning goals and different collaboration scripts. We discuss how two collaboration scripts that we are using in our research on fractions learning are implemented in CTAT. The examples illustrate how CTAT flexibly supports collaborative tutors by running synchronized tutor engines for each student, and how it supports the development of collaborative tutors through the use of multiple behavior graphs that use no programming to develop.

Keywords: Problem solving, collaborative learning, intelligent tutoring system, authoring tools

1 Introduction

Collaborative learning has been shown to be effective for student's knowledge acquisition in some computer-supported settings [9]. However, there is a lack of effective and flexible authoring tools for collaborative learning activities. Authoring tools for Intelligent Tutoring Systems (ITSs) are often geared towards *individual* learning and typically do not have support for the components that make collaborative learning effective [11]. Within Computer Supported Collaborative Learning, collaboration scripts are often used to support collaborative learning, but are often either developed specifically for a particular application [8] [17] or, at best, are provided through a tool that can be used for reuse of the *same* script across multiple subject areas [1], [3], [7], [10], [13-14], [16]. In both approaches, the development tailored for particular domains and learning goals is not straightforward and may not even be feasible. A tool that can be used to flexibly author a range of collaboration scripts for a range of subject areas would bridge this gap. We are working on creating such a tool, by extending an existing ITS authoring tool, the Cognitive Tutor Authoring Tools (CTAT) [2], so it aids in the development of tutors that integrate a range of collaboration scripts. An earlier attempt to extend CTAT [4] focused on log data, not scripting.

Collaboration scripts are used to structure the tasks and interactions within a group. According to Kollar, Fischer, and Hesse [6], a collaboration script within the educational domain consists of at least five components: the learning objectives, the types of learning activities, the sequencing of the activities, role distribution, and how the script is represented. These components are a way to compare collaboration scripts across platforms, such as face-to-face and computer-supported settings and provide a guideline for the coverage that is needed in authoring tools that wish to support collaborative learning.

There has been work to make collaboration scripts generalizable across learning domains. One example of an authoring tool that can be used across different learning domains is the work done with conversational agents, which monitor a group conversation and can intervene when needed [1], [7]. Although this authoring tool supports multiple learning domains, it supports only the development of collaboration scripts that rely on the use of conversational agents and not a more general class of collaboration scripts. Other tools aim to reuse existing collaboration scripts for new scenarios [3], [10], [13], [16]. These tools are dependent on the learning goals that the existing collaboration script supports instead of customizing the collaboration script for the desired learning goals. On the other hand, the tool, XSS, which is a framework for rapidly developing computer-supported collaboration scripts for new technologies, does support the creation of collaboration scripts to meet specific learning goals [14]. However, XSS does not have support for authoring scripts through an interface, so it may be difficult for users with less programming experience.

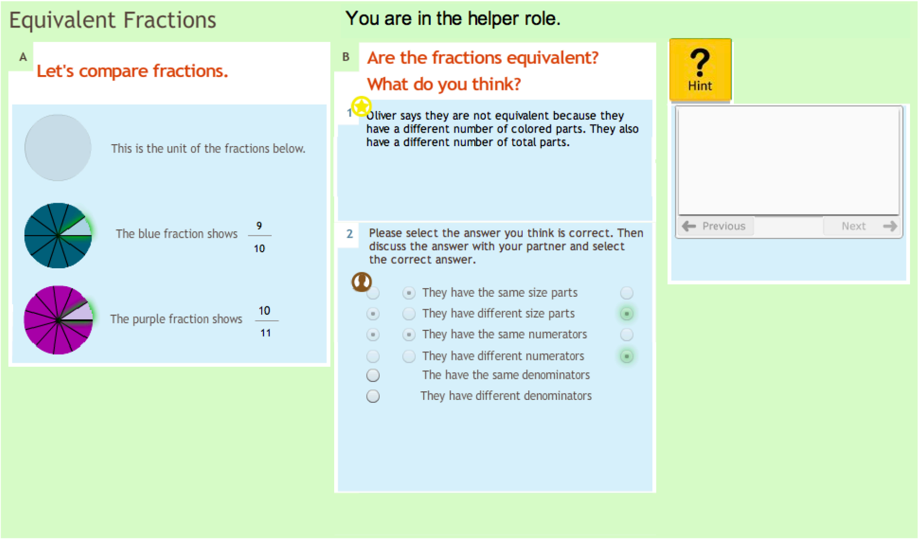
The enhancement to CTAT described in this paper allows authoring of collaborative ITSs without programming, and the collaboration script can be specific to the learning goals of the tutor being developed. In this paper we provide collaboration script examples that support cognitive group awareness [4] and sharing of unique information, illustrating the flexibility of the CTAT authoring tool for collaboration. The enhancement to the CTAT system allow students to collaborate through synchronized tutor engines and we will describe how it supports collaborative tutor problems.

2 Collaboration Examples Using CTAT for Collaboration

2.1 An Example of Support for Cognitive Group Awareness

Before we describe how we modified CTAT so it supports authoring of collaborative tutoring, we describe two examples of collaborative tutoring behavior authored with this tool. Specifically, building on our prior work on the Fractions Tutor [12], we are creating a collaborative tutoring system to help elementary students learn fractions. The current prototype includes four conceptual problems and four procedural problems focused on equivalent fractions, each with embedded collaboration scripts. The prototype tutor has been pilot tested with four dyads so far. As students use the tutor, they talk to each other via Skype. The two examples illustrate the types of collaboration scripts can be implemented using the collaborative version of CTAT. In the next section, we extended CTAT to support the collaborative features of these tutors.

The first example features a collaborative fractions problem with a script that supports cognitive group awareness, in which the student is learning conceptual knowledge about equivalent fractions. Cognitive group awareness is the awareness that comes from having information about group members' knowledge, information, or opinions and has been shown to be effective for the collaboration process [5]. This awareness can be supported through tools such as skill meters or by using an interactive interface to display a partner's answers. In our tutor, cognitive group awareness during problem solving is structured as follows: First, the collaborating partners each answer the same question separately. The tutor then displays both partners' answers to promote discussion, and the partners provide a final answer endorsed by both. Each student is given a pair of contrasting attributes (see Figure 1, panel B2) about the fractions. The students are not given feedback on their individual answer but are shown what their partner selected. This allows each student to see their partner's understanding of the fractions. The students are then asked to discuss their answers and decide as a pair what the correct answer will be. Having each student display his or her knowledge of the given fractions before discussing the question together supports the cognitive group awareness. This discussion can lead to a mutual understanding of the fraction attributes, which supports a better understanding of the conceptual knowledge for equivalent fractions. As may be clear, to support cognitive group awareness, the collaborative tutor provides different views of the same problem to the collaborating partners, using two synchronized tutor engines as described below.



Equivalent Fractions

You are in the helper role.

A Let's compare fractions.

This is the unit of the fractions below.

The blue fraction shows $\frac{9}{10}$

The purple fraction shows $\frac{10}{11}$

B Are the fractions equivalent? What do you think?

1 Oliver says they are not equivalent because they have a different number of colored parts. They also have a different number of total parts.

2 Please select the answer you think is correct. Then discuss the answer with your partner and select the correct answer.

○ ● They have the same size parts ○

● ○ They have different size parts ●

● ● They have the same numerators ○

○ ○ They have different numerators ●

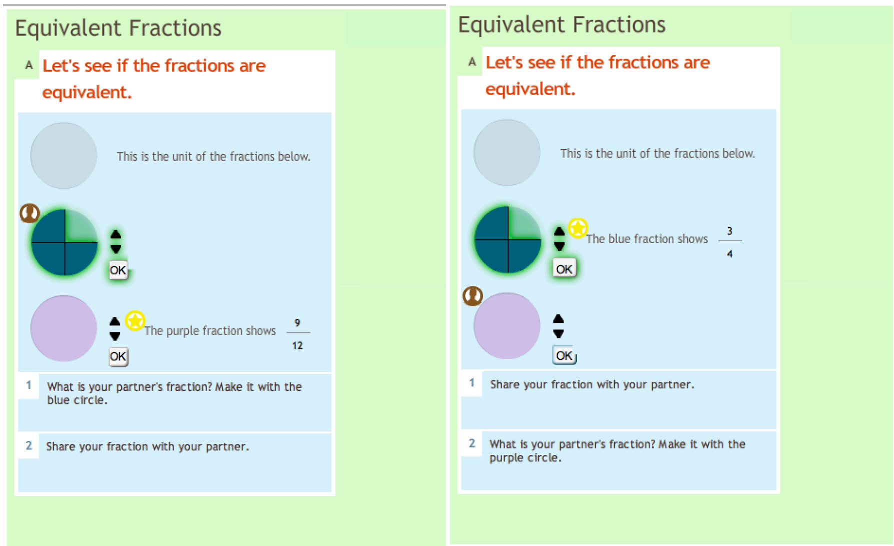
○ The have the same denominators

○ They have different denominators

Fig. 1. Panel B2 displays an example of support for cognitive group awareness through the use of multiple radio buttons where each student first selects an answer based on their knowledge before the group makes a group selection that is tutored.

2.2 An Example of Support for Sharing Unique Information

We also used the collaborative version of CTAT to implement a second type of fractions problem, in which students learn how to procedurally evaluate equivalent fractions. As in the previous example, the collaborative tutor provides a different view on the same problem for each collaborating partner, although this time the collaboration is scripted differently for the different learning objective. Specifically, we implemented a script that distributes unique information between the partners and supports the sharing of this information. Students are shown a fraction expressed in symbols (see Figure 2) that their partner does not see as indicated by the star icon. Each partner is also given a circle diagram that they can interact with; their partner can see this diagram but cannot interact with it as indicated by the silhouette icon. One student is first asked to share their fraction with their partner (i.e., by telling their partner about it) while the second student is asked to make this fraction using their circle diagram. The students then switch roles and one student shares their fraction while the other student makes this fraction. Each student sees the feedback from the tutor, so if a student is struggling to correctly make the fraction, their partner, who can see the fraction and the tutor feedback, can provide support and help. By providing each student with different information, the students need to start a dialogue and share. This activity makes the students aware of the fractions as a first step to supporting procedural knowledge for evaluating equivalent fractions.



Equivalent Fractions

A Let's see if the fractions are equivalent.

This is the unit of the fractions below.

The purple fraction shows 9/12

1 What is your partner's fraction? Make it with the blue circle.

2 Share your fraction with your partner.

Equivalent Fractions

A Let's see if the fractions are equivalent.

This is the unit of the fractions below.

The blue fraction shows 3/4

1 Share your fraction with your partner.

2 What is your partner's fraction? Make it with the purple circle.

Fig. 2. Panel A displays an example of individual information that needs to be shared between participants. The top blue fraction was made by the student on the left screen using the information shared by the student on the right screen. The purple fraction will be made with the student on the right screen with the information from the student on the left screen.

Both examples illustrate a range of collaborative activities that can be supported using CTAT for collaboration. Kollar, Fischer, and Hesse specify collaboration scripts by focusing on five components [6]. These five attributes provide a guideline for the coverage that is needed in authoring tools. Both examples use different *learning activities* to support the learning goals of the problems. The sharing of unique information uses activities such as sharing and problem solving where as the script that supports cognitive group awareness uses activities such as sharing knowledge and mutual explanations. Within these activities the students are also assigned to very different *roles* where in the unique information scenario they are asked to be a sharer or to be a problem solver and then switch roles. In the support for cognitive group awareness, both students are responsible for sharing their knowledge and then discussing the answers.

3 Authoring Tool Extensions to Support Collaboration

Until recently CTAT only supported tutors for individual use. We focus on one type of tutor that can be authored with CTAT, namely, example-tracing tutors [2]. To develop such a tutor, an author creates two key components, both without programming: a user interface designed specifically for the problem type being tutored (the interface lays out the problem steps) and a generalized behavior graph, which stores all of the acceptable solution paths along with commonly-occurring incorrect steps. The tutor uses the behavior graph to monitor student problem solving and provide guidance to students. Each behavior graphs consists of a set of links that correspond to steps that can be taken in the problem, such as typing in the numerator to a fraction. Some steps (explicitly marked as such) represent *tutor-performed actions*, such as showing a component in the tutor interface that was hidden before. To evaluate student input, the tutor compares the student's problem-solving steps against those in the behavior graph, testing whether the student is on one of the paths in the graph. An author may specify constraints on the order of steps. Behaviorally, example-tracing tutors are similar to other types of ITSs, providing all the key functionality singled out by VanLehn [15] as typical of ITSs.

3.1 Authoring Collaborative Tutors

To expand CTAT so it supports *collaborative example-tracing tutors*, we added the capability to run *multiple synchronized tutor engines,* one for each student in a collaborating group. This set up allows for great flexibility in authoring tutors with embedded collaboration scripts. Specifically, each student in a group has their own behavior graph file and interface file for the given problem. The collaborative version of CTAT synchronizes the tutors so that when one of the collaborating students takes an action, this input is sent to both that student's tutor engine and their partner's tutor engine. Similarly, tutor output is shared among the members of a collaborating group (i.e., all output from the two synched tutor engines, such as hints and feedback, is sent to each student interface separately). One result of this output sharing is that student actions taken on one interface will be "mirrored" on the other interface in the corresponding interface component, together with the associated tutor feedback. As we extended CTAT, we updated the interface tool components to include new actions that better support collaborative learning activities. As an example, we updated the existing components to allow students to view the options of a component without being able to take action on the component, as illustrated in the examples above. We are also adding a highlighting functionality so each student can easily reference a component.

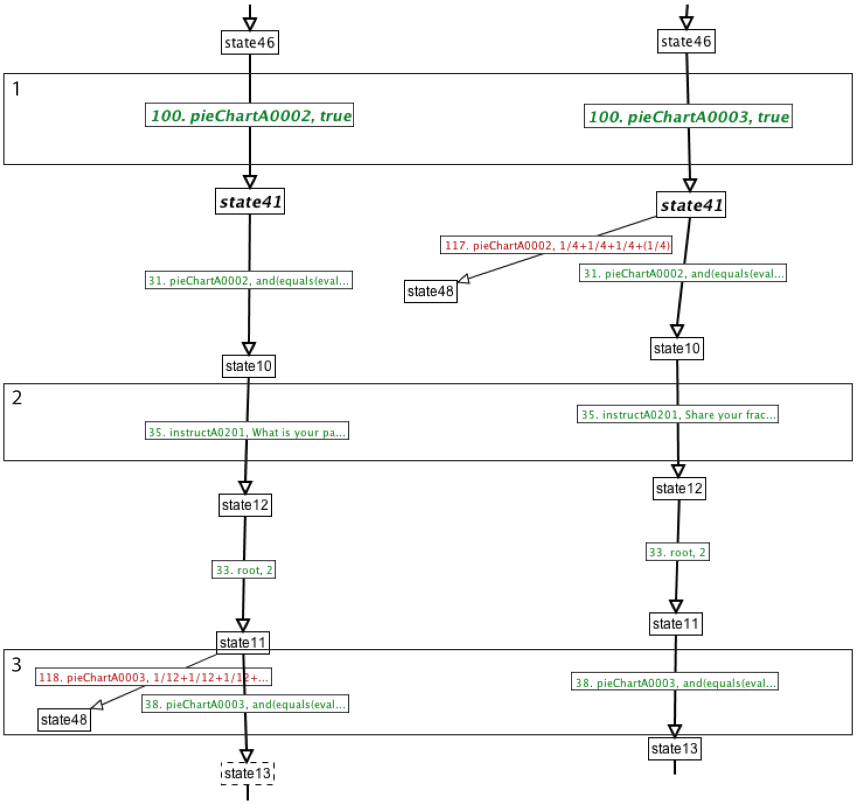

state46
state46
1
100. pieChartA0002, true
100. pieChartA0003, true
state41
state41
117. pieChartA0002, 1/4+1/4+1/4+(1/4)
31. pieChartA0002, and(equals(eval...
state48
31. pieChartA0002, and(equals(eval...
state10
state10
2
35. instructA0201, Share your frac...
35. instructA0201, What is your pa...
state12
state12
33. root, 2
33. root, 2
state11
state11
3
118. pieChartA0003, 1/12+1/12+1/12+...
38. pieChartA0003, and(equals(eval...
38. pieChartA0003, and(equals(eval...
state48
state13
state13

Fig. 3. Excerpts from two behavior graphs working for a single problem. Together both behavior graphs capture the first step to be completed by the students for the problem in Fig 2. Box 1 demonstrates the different locking of components for each student, Box 2 demonstrates different instructions for each student, and Box 3 demonstrates the use of student-performed actions to advance the state of the problem where the partnering student can only take the action.

With these collaborative extensions to CTAT, an author can create tutors that do not differ for the collaborating partners - simply by supplying the same behavior graph and interface for each collaborating partner. The result would be a tutor with which two students interact simultaneously and synchronously while each sitting at their own computer. They would each see the changes that their partner makes. This kind of collaboration may not be terribly useful, however. The power of the approach comes from being able to craft tutors in which the collaborators have different views on the same problem and have different sets of actions available to them. There are many collaboration activities, such as the jigsaw and the tutee/tutor paradigm, where the benefit of the activity comes from the students having different roles and responsibilities in the problem-solving task. The CTAT authoring tool supports this kind of differentiation, as an author can create separate behavior graphs, one for each student, that display different instructions or capture different student problem-solving actions, dependent on the role of each student, as is used in the cognitive awareness activity. For example, Figure 3 shows, side-by-side, two behavior graphs for the support of unique information example illustrated in Figure 2. These two behavior graphs share common structure, but also differ so as to support different interactions for the two collaborating students.

To show different instructions for each student, an author can use a different tutor-performed action at the corresponding link in the two behavior graphs. An example is shown in Box 2 of Figure 3 where each student receives different directions from the behavior graph at the same point in time. (The label on the link shows the message displayed to the student in truncated form.) Similarly, by providing different behavior graphs for each member, the actions taken by the users can differ. One way to make different sets of actions available to each collaborating partner is by locking certain components in the interface, a different set for each partner. This allows both students to see the action on their respective interfaces while only allowing one student to be able to take the action. An example is shown in Box 1 of Figure 1 where different components (the two circle components, pieChartA0002 and pieChartA0003) are locked for the students through a tutor-performed action, preventing them from interacting with that component. The result of this link in the behavior graph is seen in Figure 2 where the circle that corresponds to the fraction shown on the screen is locked for that student, so that each student can perform his/her own role but not his/her partner's role. Though the student cannot act on the component that is locked, a step to solve the component is in the behavior graph (see Box 3 of Figure 3) so that the problem will not advance until their partner has completed the step. An author can also make the tutor accept different actions from each student by recording different actions in each student's behavior graph. In this case, the student without the action recorded would not have to wait for this action to take place to continue working on the next step of the problem.

Another way to provide different interface elements to the members of each dyad is through an interface file. This file is a SWF file created in Flash. The author can select the components, control their placement on the interface, set basic parameters, and use custom code if necessary. In this way, an author can tailor the interface for the different roles that the collaborators have in the collaboration script that is being supported. An author can also determine what feedback each student receives during the problem by setting an initial tutor feedback parameter for each interface component. This parameter controls whether or not there will be tutor feedback on actions on that component. For example, in the cognitive awareness task in Figure 1, the radio buttons that correspond to the student's *individual* answers provide no feedback, as they serve mainly to support the partners' mutual awareness of each other's reasoning. On the other hand, the radio buttons for the *group* answer (on the right in Figure 1) provide correct or incorrect feedback.

The steps to develop a tutor using CTAT consist of developing a user interface, creating a behavior graph, and annotating the behavior graph [2]. Within CTAT, an interface is built using an interface builder and the different components of the interface are adding using a drag-and-drop method. Each component has a set of parameters that can be set allowing the developer to customize the look and feel of the parameter to match their tutor layout. This allows a developer to create a tutor interface without the need for any coding on the part of the developer. Once an interface is created, a behavior graph can be created that maps out the tutor steps through correct and incorrect actions. The behavior graph can be created through demonstrating the actions to be taken on the interface. While having the CTAT Behavior Recorder in demonstration mode, any action that is taken on the interface will be recorded on a behavior graph. By starting at different points in the behavior graph, different branches can be created. This allows a developer to create a behavior graph without the need of programming. After the behavior graph is created through demonstration, the graph can be annotated. Annotation includes adding hints to the links and identifying knowledge components.

To author a collaborative tutor each of the steps to create an individual tutor are followed for each member of the collaboration. Depending on the type of collaboration activities and roles depends on if different tutor interfaces and behavior graphs need to be made for each student in the group or if the same files can be used. If the students are going to be seeing something visually similar then the same tutor interface can be used. If the students are going to take the same actions during the problem, then the same behavior graph can be used. When developing a collaborative tutor, if different interfaces are going to be used and an action that one student takes should be reflected in the view to the other students, then the components that are used for this activity need to be named the same in both interfaces. This is shown in Box 3 of Figure 3 where the same component name is referenced in both behavior graphs. This allows the tutors to reflect an action taken on one interface on the other interface as well. On the other hand, if the author wants particular actions within a tutor interface to be private to one of the collaborating students, one way to do so is to not provide a corresponding interface component in the interface for the other student. The enhancements to CTAT did not add a need for a developer to program to create a tutor. Currently, to test a collaborative tutor, the tutor must be run through the tutoring service. A different browser window can be opened for each student interface so the actions can be seen simultaneously. By assigning each interface and behavior graph to a "student" and then identifying those students as being in a class together, the different tutors are synced and allows communication between the tutors. This assignment can be done through filling out fields in a user interface and no special programming is needed on the part of the author.

4 Discussion, Conclusion and Future Work

Computer-supported collaboration has been shown to be an effective learning paradigm for knowledge acquisition [9], yet most tools that support collaboration do not allow for the authoring of a range of collaboration scripts. Authoring tools for ITSs have been used to address a wide range of domains, but we are not aware of any that support collaboration scripts, other than an early attempt to extend CTAT [4] so it builds collaborative tutors from log data. We extended CTAT so it supports the authoring of collaborative tutors while maintaining its advantages for individual tutoring without programming. With this new version of CTAT, authors can develop collaborative ITSs to meet a range of domains and collaboration scripts. The developer does not need to have a strong background in programming to make a functional tutor.

The extension to CTAT allows for a range of collaboration scripts to be developed. Two examples were provided in this paper, but we also created tailored collaboration scripts to match the learning objectives of six other fractions problems. The flexibility to develop these scripts is because the collaborative version of CTAT was not developed to implement a specific script but to remain open-ended. This design also allows flexibility while developing tutors. As we continue to develop our collaborative fractions tutor, we are taking an iterative approach in which we repeatedly test the collaboration script with students and then refine it to best support the learning goals based on the outcomes of the pilot studies. The collaborative version of CTAT allows for these changes to be made easily in a problem, as behavior graphs are relatively malleable.

Future work will consist of extending CTAT so it can support more than two students in the group. Other future work will be to allow the specifying of groups at runtime instead of needing to specify groups ahead of time. By being able to specify the members of a group at runtime, there would be more flexibility in grouping students in a classroom on any given day. Students would not be dependent on their partner also being there that day. Also to improve the authoring process, functionality is being added to allow an author to have multiple behavior graphs open so they can compare the steps and can copy and paste steps from one graph to another that are similar. The eventual goal of our project is to investigate how best to combine individual and collaborative modes of learning.

Acknowledgments. We thank the Cognitive Tutor Authoring Tools team for their help. This work was supported by Graduate Training Grant # R305B090023 and by Award # R305A120734 both from the US Department of Education (IES).

5 References

1. Adamson, D., & Rosé, C. P.: Coordinating Multi-Dimensional Support in Collaborative Conversational Agents. In: Intelligent Tutoring Systems (pp. 346-351). Springer Berlin Heidelberg (2012)

2. Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K. R.: A New Paradigm for Intelligent Tutoring Systems: Example-tracing Tutors. International Journal of Artificial Intelligence in Education, 19, 105-154 (2009)
3. Harrer, A., Malzahn, N., & Wichmann, A.: The remote Control Approach-An architecture for Adaptive Scripting Across Collaborative Learning Environments. Journal of Universal Computer Science, 14(1), 148-173 (2008)
4. Harrer, A., McLaren, B. M., Walker, E., Bollen, L., & Sewall, J.: Creating cognitive tutors for collaborative learning: Steps toward realization. User Modeling and User-Adapted Interaction, 16(3-4), 175-209 (2006)
5. Janssen, J., & Bodemer, D.: Coordinated Computer-Supported Collaborative Learning: Awareness and Awareness Tools. Educational Psychologist, 48(1), 40-55 (2013)
6. Kollar, I., Fischer, F., & Hesse, F. W.: Collaboration scripts–a conceptual analysis. Educational Psychology Review, 18(2), 159-185 (2006)
7. Kumar, R., Rosé, C. P., Wang, Y., Joshi, M., & Robinson, A.: Tutorial dialogue as adaptive collaborative learning support. Frontiers in Artificial Intelligence and Applications, 158, 383 (2007)
8. Lesgold, A., Katz, S., Greenberg, L., Hughes, E., & Eggan, G.: Extensions of intelligent tutoring paradigms to support collaborative learning. In S. Dijkstra, H. P. M. Krammer, & J. J. G. van Merrienboer (Eds.), Instructional models in computer-based learning environments. (pp. 291-311). Berlin: Springer-Verlag (1992)
9. Lou, Y., Abrami, P. C., & d'Apollonia, S.: Small group and individual learning with technology: A meta-analysis. Review of educational research, 71(3), 449-521 (2001)
10. Miao, Y., Hoeksema, K., Hoppe, H. U., & Harrer, A.: CSCL Scripts: Modelling Features and Potential Use. In Proceedings of the 2005 conference on Computer support for collaborative learning: learning 2005: the next 10 years! (pp. 423-432). ISLS (2005)
11. Murray, T., Blessing, S., & Ainsworth, S.: Authoring tools for advanced technology learning environments: Toward cost-effective adaptive, interactive and intelligent educational software. Amsterdam: Kluwer Academic Publishers (2003)
12. Rau, M., Aleven, V., Rummel, N., & Rohrbach, S.: Sense Making Alone Doesn't Do It: Fluency Matters Too! ITS Support for Robust Learning with Multiple Representations. In: Intelligent Tutoring Systems, pp. 174-184. Springer Berlin/Heidelberg (2012)
13. Ronen, M., Kohen-Vacs, D., & Raz-Fogel, N.: Adopt & Adapt: Structuring, Sharing and Reusing Asynchronous Collaborative Pedagogy. In Proceedings of the 7th International Conference on Learning Sciences (pp. 599-605). ISLS (2006)
14. Stegmann, K., Streng, S., Halbinger, M., Koch, J., Fischer, F., & Hußmann, H.: eXtremely Simple Scripting (XSS): A framework to speed up the development of computer-supported collaboration scripts. In Proceedings of the 9th International Conference on Computer supported Collaborative Learning-Volume 2 (pp. 195-197). ISLS (2009)
15. VanLehn, K.: The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. Educational Psychologist, 46, 197-221, (2011)
16. Wecker, C., Stegmann, K., Bernstein, F., Huber, M. J., Kalus, G., Rathmayer, S., Kollar, I., & Fischer, F.: Sustainable script and scaffold development for collaboration on varying web content: the S-COL technological approach. In Proceedings of the 9th international conference on Computer supported collaborative learning-Volume 1 (pp. 512-516). ISLS (2009)
17. Walker, E., Rummel, N., & Koedinger, K.: CTRL: A research framework for providing adaptive collaborative learning support. User Modeling and User-Adapted Interaction: The Journal of Personalization Research (UMUAI), 19(5), 387-431 (2009)

A Data Mining Approach to Construct Production Rules in an Educational Game

Fazel Keshtkar, Borham Samei, Brent Morgan, and Arthur C. Graesser

University of Memphis
Institute for Intelligent Systems
Memphis, TN, USA.
{fkshtkar,bsamei,brent.morgan,a-graesser}@memphis.edu

Abstract. One of the most crucial aspects of Intelligent Tutoring Systems in a collaborative serious game is production rules. Given the large number of interactions and conversation between players, it is difficult to follow student questions and reactions in the game environment. Therefore, creating a sophisticated method to construct production rules for handle the students' interactions will boost the performance of the system. In this paper, we propose a state-of-the-art computational approach to automatically generate production rules using co-occurrences of distinct terms from a corpus of students' conversations. Moreover, our model is able to generate additional production rules as new data is available. Finally, we also introduce how to transfer extracted co-occurrences into production rules, and how to build these into the game system.

Keywords: Intelligent Tutoring Systems, Production Rules, Data Mining

1 Introduction

Serious games are increasingly becoming a popular, effective supplement to standard classroom instruction [9]. Some classes of serious games provide microwords [7] that allow players to explore a virtual environment. These simulations have ideal and often simple problems with targeted scaffolding to help users identify important concepts and think critically about them. Multi-party chat is pervasive in recreational games and often crucial to success in multi-player epistemic games [4, 3, 8]. In this paper, we present a method of production rule We employed a computational approach to determine the critical features of multiparty chat in a serious game. We analyzed a corpus of chat conversations and high frequency features along with their co-occurrences. We describe the resulting model below, as well as the process of generating production rules. Finally, we discuss how to utilize this model in the context of a serious game to provide relevant suggestions to a human mentor.

2 Production Rules

A Production Rule consists of a collection of *IF...THEN* rules that together form an information processing model of some task, or range of tasks. Each rule has two parts: a condition part and an action part. Production rules can be represented in various forms [2], e.g.: “IF condition THEN action”, “IF premise THEN conclusion” or on the other hand “IF proposition p_1 and proposition p_2 are true THEN proposition p_3 is true”. In the context of a serious game, for example, it is likely that the players will eventually need help navigating the user interface. Whereas they would normally ask a human mentor to guide them, if a relevant production rule is built in the system, this situation can easily be detected and resolved by the system, saving the resources of the human mentor. The system outlined below must be able to detect the specific facts or features (such as “email” and “check”) to specify relevant conditions and return the appropriate suggestion. As a result, a computational data mining approach helped us to extract these conditions and facts.

2.1 Speech Act Classification

We selected a system for classifying speech acts [5]. Analyzes of a variety of corpora, including chat and multiparty games, have converged on a set of speech act categories that are both theoretically justified and that also can be reliably coded by trained judges [6]. Our classification scheme has 8 broad categories: **Statements** are verbal reports on scientific facts. **Requests** include asking other participants in the conversation to provide information. **Questions** are queries for information from the addressee. **Reactions** are short verbal responses to requests or questions. **Expressive Evaluations** consist of feedback regarding the player’s performance. **MetaStatements** are statements about the communication process. **Greetings** are expressions regarding any party’s entrance to. **Other** represents speech acts which did not fit into the above categories.

2.2 Land Science Game

Urban Science is an epistemic game created by education researchers at the University of Wisconsin-Madison, designed to simulate an urban planning practicum experience [1]. Young people role-play as professional urban planners in an ecologically-rich neighborhood. The players’ primary task is to redesign the city of Lowell, Massachusetts. Players are assigned to one of three planning teams, and interact with team members and a human mentor using a group chat interface [4, 3, 8]. The ”Question” category is likely the most critical speech act when it comes to addressing player problems. We analyzed 26720 unique chat turns across three instances of Land Science data set.

3 Our Approach

In our model, we identify the relevant facts needed to satisfy the conditions in *IF ... THEN*. Based on these facts, we are able to generate suggestions for a human mentor. In our algorithm, facts can have any of the following features: words, tokens, event, status of the game, or patterns of player’s conversation.

Table 1. shows some of tokens that have high co-occurrence

Token 1	Token 2	co-occ	categories	rooms
stakeholders	what	413	Statement Question Request Reaction	7 5 3 4 12
email	maggie	353	Statement Request Question ExpressiveEval	3 2 4 5 6
want	what	306	Statement Request Question Reaction	5 7 3 4 12
meeting	team	281	Statement Request Question ExpressiveEval	4 3 2 10 11
out	what	280	Statement Request Question Reaction	7 4 3 5 2
now	what	262	Statement Question Request Reaction	7 3 2 10 1
find	out	257	Statement Question Reaction Request	5 10 4 3 7
final	proposal	237	Statement Reaction Request Question	12 2 11 3 6
preference	survey	236	Statement Request Reaction Question	6 9 7 5 10
stakeholders	want	229	Statement Request Question Reaction	5 7 4 3 12

Using these facts, we can generate production rules which offer suggestions for a human mentor.

3.1 Computing Co-Occurrences

One of the most important features to build production rules based on a data-mining approach is to determine the co-occurrences of high or even low frequency tokens in the corpus. In the following sections we describe these features and we show how they can be considered as conditions and facts in our production rules. After preprocessing the corpus, we split each utterance into tokens using the OpenNLP tokenizer, a Natural Language Processing Java Library. We used standard stop words to remove unnecessary tokens. We computed the frequency of all remaining tokens in the corpus for each Speech Act category. These tokens are based on Unigram Entropy Cues and Speech Act classification method that developed by [5]. Then, we ranked these frequencies list from high to low order. In addition to token frequency, it is also critical to assess the relevance of each token, as it may be context-specific. We assessed token relevance by computing co-occurrences. Table 1 shows some examples of co-occurrences in our corpus. In Table 1 tokens that have high co-occurrence chance along with the categories and rooms they appeared in. The categories and rooms are ordered by the frequency of the co-occurrence.

3.2 Constructing Production Rules

As we described in previous sections, Production Rules are in forms of IF ... $THEN$ statements. These IF ... $THEN$ statements must obtained by the **Conditions** and the **Facts** to achieve some **Conclusions** or **Actions**. By looking at Table 1, we see the co-occurrences for “Virtual” are: navigation, stakeholder, neighborhood, character, site, during, etc. In our model, we assume that the facts for conditions can be one or more of the co-occurrences for each token.

4 Conclusion and Future Work

In this paper, we discussed the concept of production rules. These rules are $IF...THEN$ statements which contain some conditions (based on relevant facts). When conditions are met, they trigger some system response, such as a suggestion to players from a mentor or intelligent agent. We introduced a state-of-the-art data-mining approach to construct production rules from a corpus of chat conversations. For future work, we plan to use rule based model to generate production rule. This will allow us to fire relevant functions to produce better suggestions. We also plan to analyze more data to construct additional production rules for the Land Science.

Acknowledgments

This work was funded by the National Science Foundation (DRK-12-0918409).

References

1. Bagley, E.: Stop Talking and Type: Mentoring in a Virtual and Face-to-Face Environmental Education Environment. Ph.D. thesis, University of Wisconsin-Madison (2011)
2. Compton, P., J.R.: A philosophical basis for knowledge acquisition. knowledge acquisition 1990; 2: 241-257
3. D. Ketelhut, C. Dede, J.C.B.N.C.B.: Studying situated learning in a multi-user virtual environment. In: Baker, E., Dickieson, J., Wul-feck, W., ONeil, H. (eds.) Assessment of problem solving using simulations, pp. 37-58. Earlbaum, Mahweh (2007)
4. Dieterle, E., Clarke, J.: Multiuser virtual environments for teaching and learning. In: Pagani, M. (ed.) Encyclopedia of multimedia technology and networking (2nd ed). Idea Group, Hershey 2012.
5. Moldovan, C., Rus, V., Graesser, A.C.: Automated speech act classification for online chat. In: Procceding of FLAIRS 2011 (2011)
6. R.G. DAndrade, M.W.: Speech act theory in quantitative research on inter-personal behavior. In: Discourse Processes. 8 (2), 229-259 (1985)
7. Rickel, J., Lesh, N., Rich, C., Sidner, C., Gertner, A.: Building a bridge between intelligent tutoring and collaborative dialogue systems. In: Proceedings of the 10th Int. Conf. on Artificial Intelligence in Education, San Antonio, TX, pp. 592-594, May 2001.
8. Shaffer, D.: How Computer Games Help Children Learn. Palgrave, New York (2007)
9. Shaffer, D., Chesler, N., Arastoopour, G., DAngelo, S.: Nephrotex: Teaching first year students how to think like engineers. Laboratory Improvement (CCLI) PI Conference (2011)

Modeling the Process of Online Q&A Discussions using a Dialogue State Model

Shitian Shen and Jihie Kim

University of Southern California Information Sciences Institute

4676 Admiralty Way, Marina del Rey, CA, U.S.A

shitians@usc.edu, jihie@isi.edu

Abstract:

Online discussion board has become increasingly popular in higher ed-ucation. As a step towards analyzing the role that students and instructors play during the discussion process and assessing students' learning from discussions, we model different types of contributions made by instructors and students with a dialogue-state model. By analyzing frequent Q&A discussion patterns, we have developed a graphic model of dialogue states that captures the information role that each message plays, and used the model in analyzing student discussions. We present several viable approaches including CRF, SVM, and decision tree for the state classification. Using the state information, we analyze information exchange patterns and resolvedness of the discussion. Such analyses can give us a new insight on how students interact in online discussions and kind of assistance needed by the students.

Keywords: online discussions, dialogue transition, speech act, CRF.

1. Introduction

Online discussion boards, an application of social network on education, provides a platform for students and instructors to share their ideas or to discuss their question not only in traditional courses but also in web-based courses. Such tools can help students solve their problems opportunely, as well as improving instructors' work efficiency. As the discussion board usage increases, we want to understand how students interact with instructors and peers, and how they learn through that interaction.

Although research in online chat and discussion analysis has been increasing re-cently [8,12,14], there has been limited research on modeling the process of information exchange in Q&A forums or how resolvedness of discussions can be determined. In order to analyze and model the process of information exchange, we map interactions in discussions into a Q&A dialogue state model. The state for each message illustrates the status and function of the given message in the Q&A process (discussion thread) [5,6]. We identified six distinctive and frequent states in the discussion process: Problem presenting, Problem understanding, Solving, Solution understanding, Solution objecting, and Solution appreciation. In order to classify the dialogue states efficiently, we apply machine classifiers including linear Conditional Random Fields (CRF), a widely used tool for characterizing the sequential data. The features are generated from message content and positional information, including cue word posi-tions, participants' order, which provides additional hints for state labeling.

The results indicate that frequent states can be reasonably identified using ma-chine classifiers. We demonstrate that the state model can be used in finding frequent patterns in the dialogue progress and evaluating the roles the instructor and students play during the Q&A discussion process. Furthermore, we show that state information can help identifying unresolved discussions, which can be reported to the instructor.

2. State transition model of Q&A discussions

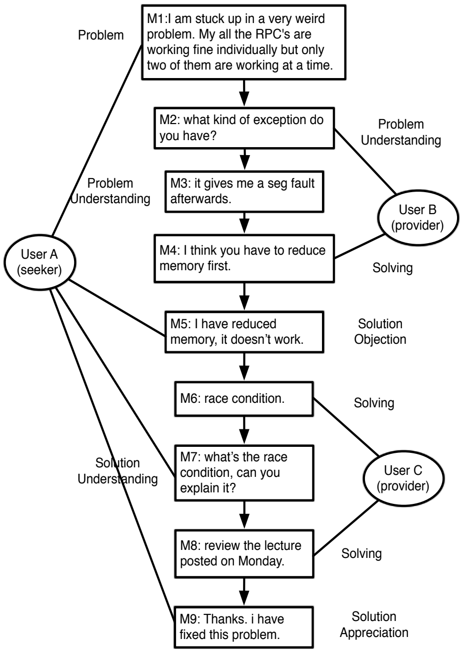

Problem
M1:I am stuck up in a very weird problem. My all the RPC's are working fine individually but only two of them are working at a time.
M2: what kind of exception do you have?
Problem Understanding
Problem Understanding
M3: it gives me a seg fault afterwards.
User B (provider)
User A (seeker)
M4: I think you have to reduce memory first.
Solving
M5: I have reduced memory, it doesn't work.
Solution Objection
M6: race condition.
Solving
Solution Understanding
M7: what's the race condition, can you explain it?
User C (provider)
M8: review the lecture posted on Monday.
Solving
M9: Thanks. i have fixed this problem.
Solution Appreciation

Fig.1. An example of discussion thread.

We use discussion corpus from undergradu-ate operating systems courses. The courses contain programming projects, and students use discussions to share problems and get help from the instructor and other students. Figure 1 shows an example discussion thread with a sequence of message. User A, B and C represents the participants. User A initiates the thread by describing the problem and asks for help. User B asks for more details related to the problem and User A provides some information. User B then gives a possible solution and User A complains that it doesn't solve the problem. User C offers another answer, and User A asks a related question. User C provides an additional suggestion. Finally, User A acknowledges the help with thanking.

Through analyses of the discussion corpus, we identified six distinctive and fre-quent states. User roles are relevant to characterizing the states: information seeker and information provider, and often the role of a user stays the same within a short dis-cussion thread [16]. The first state (Problem or P) is presented by a Seeker. In Figure 1, M1 can be regarded as a P state. In the second state (Problem Understanding or PU), the problem is further elaborated and discussed. PU can consist of multiple messages. Another discussant (student or the instructor) may post a question in order to under-stand the problem that the seeker confronts. Such questions are usually followed by an answer by the seeker who posted the problem. For example, M2 and M3 help the participants understand the problem. In the third state (Solving or S), a participant provides a direct solution or a hint. Although we label it as S, the grammatical form for such messages may vary. For example, hints can be provided as a question:”why not try ABC?”. After Solving, the seeker (or other participants) can respond with Solution Appreciation (SA), Solution Objection (SO) or Solution Understanding (SU). In SA, seeker can acknowledge the assistance with thanking, like M9.

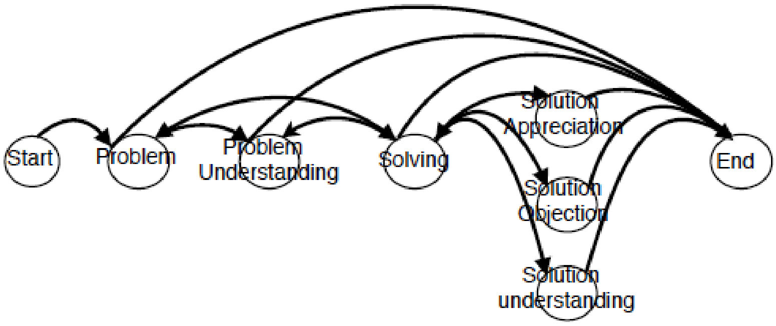


Fig. 2. State transition model for Q&A discussions

Table 1. A Q&A State Model: Definitions and Examples.

State	Definition	Example	Count	Kappa
Problem (P)	Original problem is proposed by information seeker	I stuck in a very weird problem…..	251	0.98
Problem Understanding (PU)	1.Providers ask related questions for understanding original question 2. Seekers answer the related questions and supply more details related to original issues.	1.What kind of exception do you have? 2. It’s seg fault afterwards	49	0.96
Solving (S)	Information providers supply answer or suggestions for solving original question	You can try to reduce the memory	447	0.99
Solution Appreciation (SA)	Seekers solve the problem and acknowledge the help from providers	It works, Thanks.	25	0.92
Solution Objection (SO)	Seekers find the answer doesn’t work and may ask for more help.	It doesn’t work, any ideas?	18	0.88
Solution Understanding (SU)	Seekers may be confused about answer and ask questions for understanding.	What’s the race condition, can you explain it?	108	0.97

Note that not all of the messages containing the ‘thank’ words can be labeled as SA because some P messages can contain ‘thanks’ in advance before a solution is provided. In SO, the seeker or another participant objects or criticizes the answer proposed by a provider, as shown in Figure 1. SU may appear when the seeker fails to understand the solution and may ask for more information. M7 is an example. Note that it is hard to identify the difference between PU and SU only based on the content of the message because similar words may be used in both states. However, the context or the dialogue state of the message can help distinguishing the two. In Figure 2, we illustrate transitions among these states. P state can be followed by a PU as well as a S but its transition to a SA, a SO, or a SU is rare.

Table 1 presents a description of each state and examples. The state information is annotated manually and the last column shows the Kappa values for agreement between two annotators. The table also shows the distribution of the states. We can find that almost 50 percent of states belong to S. There is a small number of SOs.

Table 2. State transition matrix frequencies

state	P	PU	S	SA	SO	SU
P	-	14	220	-	-	-
PU	-	20	19	-	-	-
S	9	16	101	22	17	92
SA	-	-	4	4	-	3
SO	-	-	13	-	-	-
SU	-	-	90	-	-	10

Table 2 shows the frequency of state transitions. We can find that S is a bridge between the first two states and the last three states. The first two states (P and PU) discusses about the problem to be solved, while the last three are the feedback to the solution, and S connect the two parts. S dominates in the corpus. A S often directly follows a P, but there are cases where the Q&A process goes through a PU. Below we examine frequent patterns in the discuss process using the state information.

3. Automatic Discussion State Classification

236 threads and 899 posts are used for constructing the state transition model.

Data preprocessing, normalization, and feature generation

Student discussion data is highly noisy due to variances and informal nature of student written messages. The data pre-processing steps convert some of the informal expres-sions. For example, “yep”, “yeah” and “yea” are all substituted by “yes”. “what’s” and “wats” have to be converted to “what is”. The features for state classification are generated from (a) the message content, (b) neighboring messages, and (c) the mes-sage/author locational information:

-F1: n grams features within current message

-F2: position of the current message, such as the first message, the last message

-F3: position of participants, like the first author, the last author

-F4: n grams features within the previous message

-F5: position of the previous message

-F6: position of previous author

Table 3. Top 3 features for each state

P	PU	S	SA	SO	SU
[get] unigram103_NotFirst	[get] unigram103_NotFirst	2ndAuthor	[correct] unigram197_Bottom	1stAuthor	[fine] unigram330_NotFirst
[somehow+delet] bigram65_ Any	[somehow+delet] bigram65_ Any	[get] unigram103_NotFirst	[Cat _WH+should] bigram421_ Any	replyTo2ndMessage	[it+okai] PRE_bigram248_Bottom
2ndAuthor	[about] PRE_unigram134_ Any	[somehow+delet] bigram65_ Any	[Cat _Subj_IWE+had] bigram581_Any	[give+Cat_Objective _IWE] bigram154_NotFirst	[Cat _BE+wrong] PRE_bigram393_NotFirst

Given the full features generated from the content and the position, we use In-formation Gain [15] to reduce the features space. We select the 1620 features. The top 3 features for each state are shown in Table 3. Some of the features are n-grams from the current message or the previous message, e.g., a unigram CAT_ISSUE and a bigram not+sure. PRE represents feature from the previous message. Top/Bottom/Any/NotFirst represent position of the cue words in the message.

Linear CRF and other machine learning methods

Linear CRF [9] is a probabilistic model for characterizing the sequential data, referring the feature information, as well as the dependency among neighbors. The probability function are presented in equation (1) as follows,

$$\mathrm{p}\left({}^{Y}\!/_{X}\right) = \frac{1}{Z(x)} \exp\left\{\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t)\right\} \qquad (1)$$

where Z(x) is an instance-specific normalization function, defined as equation (2),

$$Z(X) = \sum_{y} \exp\left\{\sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x_t)\right\} \qquad (2)$$

Y is the sequence data, X is the feature vectors with the total number K.   is a parameter vector, and the corresponding feature functions are defined as .

In our application, each thread, containing several messages, is regarded as the sequence data. Linear CRF can capture the dependence among these messages, and give a most likely state transition with the purpose of characterizing each state for each message in a thread. We use Mallet [7] to create the model. Other machine learning methods such as SVM, decision tree, and logistic regression are widely used in practice. Since differences among states are rather clear and the data space is partitional, decision tree can build the model by separating feature space iteratively. SVM is also used as it is sensitive to the data points near the states' boundaries and has been suc-cessfully used for many problems. Logistic regression is another effective algorithm for categorical variables. Weka [10] was employed.

Resampling

We apply sampling methods due to unbalanced data. We split the six states as majority classes, including P, S and minority classes containing the rest four states. Because SVM, decision tree and logistic regression regard each message independently, resampling method can be applied directly by adding a copy of each minority instance.

As linear CRF rely on the message sequence, we separate threads as majority and minority classes. Majority thread can be defined as threads that have only P and/or S state, while minority threads include at least one message with other states: PU, SA, SO and SU. A combination of downsampling and upsamping methods is utilized for balancing the data and obtaining the better results; we reduce the majority threads by 30% and duplicate minority threads twice. For each classifier, we performed 10-fold cross-validation. In each fold, we separate data randomly, and use 80% for training data and 20% for test. Resampling is done for training data only.

Classification Results

Table 4. Classification Results

	Precision/Recall/F-measure (%)					
Model	**P**	**PU**	**S**	**SA**	**SO**	**SU**
Linear CRF	98.1/95.3/**96.7**	32.0/20.6/**25.0**	86.4/90.6/88.5	43.1/38.8/40.8	23.3/12.4/16.2	62.2/74.0/**67.5**
SVM	100/93.8/**96.7**	15.8/36.7/22.1	88.7/91.1/**90.0**	42.1/63.0/**53.6**	24.1/56.7/**31.2**	53.8/90.6/**67.5**
J48	99.6/94.1	10.1/28.7/15.8	83.0/89.0/85.2	22.5/48.8/29.1	10.8/23.3/14.3	47.6/80.1/59.5
LR	87.2/87.5/87.3	12.1/22.7/15.8	85.8/87.9/85.2	41.0/56.3/29.1	22.8/15.0/14.3	41.8/59.6/59.5

Table 4 shows precision, recall and F-measure scores for different classifiers. Linear CRF, SVM perform better than logistic regression and decision tree. It seems that the relation between states and features are not fully captured through a non-linear function directly. Although SVM and decision tree regard messages individually, both methods make use of dependencies among neighboring messages as some of the features capture previous message content and location information. Because of the small size for state PU, SA and SO, the precision and recall for these three states is low, especially for decision tree, which is sensitive for the features and instances. The precision and recall for state SA is relatively high. A possible reason is that its features include useful cue words including "thanks", "it works" that appear regularly. On the other hand, although we have 108 instances for state SU, the precision and recall for it is not so high. We may need further examples due to its variances. Another reason is that SU often contains a question for the solution, which may use similar key words as in P, thus it's challenging to completely distinguish SU from P.

4. Analyzing Q&A Process with State Information

Frequent dialogue patterns

We use the classified information in analyzing frequent state transitions and dialogue patterns. State transitions are represented as a sequence of three states: " previous state -> current state -> next state". We further distinguish contributions by the instructor and students. The end of discussion is labeled as "end". We list the top ten frequent patterns from 236 discussion threads in Table 5.

Table 5. The top ten frequent patterns for both instructor and students

Instructor			Student		
pattern	count	percent	pattern	count	percent
P->S->end	88	13.31%	S->SU->S	77	11.65%
P->S->SU	36	5.45%	P->S->S	33	4.99%
SU->S->end	30	4.54%	S->S->S	26	3.93%
S->S->end	20	3.03%	P->S->end	24	3.63%
SU->S->SU	17	2.57%	SU->S->S	16	2.42%
S->S->SU	12	1.82%	S->SO->S	13	1.97%
P->S->PU	8	1.21%	S->S->end	13	1.97%
PU->S->end	8	1.21%	S->S->SU	12	1.82%
S->S->S	7	1.06%	S->SU->SU	9	1.36%
SU->S->S	6	0.91%	SU->SU->S	9	1.36%

The trends include:

a. Most SUs are generated by students.

b. The most frequent pattern for instructors is “P->S->end”, and its frequency is much higher than the corresponding students’ pattern. It indicates that instructor’s answers can end many discussions, and may discourage further participation by the student.

c. If the previous state is P, most of the current states, generated either by the instructor or the students, is S. Instructor answers may be followed by SU: “P->S->SU”, In contrast, students’ S states tend to be followed by another S. That is, additional or different answers are proposed to students’ answers more often than to instructors’ answers.

d. If the previous state is SU, the instructor tends to post S, and the next state is often SU. Given a SU, students post either S or SU, and it can follow by another S.

e. If the previous state is S, students tend to post US, which is followed by a S. This is the most frequent pattern for students. Students can also post a S in response to S, which can be followed by another S. The second most frequent pattern is ”S->S->S”.

f. If the previous state is PU, the instructor tends to post a S. Students may post PU in response to a PU, which is followed by S or PU. Generally speaking, students may need more discussion turns to comprehend the problem.

Timing of responses

Table 7 lists frequent state transitions based on time information. “N/A” means that there is no such state transition in the instructor pattern. ‘Instructor’ columns represent time interval values when the current message is posted by the instructor. Likewise, ‘Student’ columns show time intervals when the current message is posted by a student. According to the Table 7, we can observe the following.

Table 7. Time interval for state transition

Previous state ->Current state	Instructor		Student	
	Median	Mean	Median	Mean
P->S	4:38:39	7:56:37	1:55:11	5:29:28
P -> PU	3:36:7	6:23:6	3:9:58	3:37:16
PU -> PU	1:37:32	1:4:21	2:16:4	8:19:21
PU -> S	4:27:53	8:16:52	0:57:45	5:44:10
S -> S	5:25:58	8:49:43	1:34:26	5:41:39
SU -> S	4:22:19	8:10:59	1:18:58	3:22:22
S -> SA	1:4:37	4:30:39	0:45:54	2:17:21
S -> SO	N/A	N/A	1:55:21	5:2:18
S -> SU	N/A	N/A	1:59:2	9:1:9

1. From P state to S state, usually students spend less time in posting S than the instructor.

2. Student will spend less time to positively acknowledge (correct) answers. In other words, SA is quickly followed by a S. Transitions from S to SA, SO, and SU take a longer time. If the answer doesn’t work, students may spend more time to check their problem and work.

3. The most time consuming state transition is when the instructor posts S in response to a S.

4. Usually, students reply messages more promptly than the instructor.

5. Resolved/Unresolved Discussion Classification

A discussion thread is ‘resolved’ when all the problems proposed by the participants, including initial problems, derived problems are solved successfully. Otherwise, it’s unresolved thread. The features used for thread classification are:

-F1: n gram features within the final message in a thread

-F2: position of the final message, such as the first message, the last message

-F3: position of the final author, (the first author, the last author)

-F4: n gram features within the previous message

-F5: position of previous message and previous poster

-F6: state information

Table 8 presents the thread classification result. Comparing the three tables, we can conclude that state information indeed improve the performance of classifiers for thread classification. The state information represents the role of the message and effectively abstract low-level feature content or locational features. The state infor-mation also captures the dependencies among the messages within the whole thread, and can provide additional context information. For example, if the last state is PU, without state information, the message can be labeled as S for the understanding problems, and the classifier may label it as the resolved because it provides a solution. Generally, such abstractions provide better performance in machine classification when training data is not enough [15]. They also assist human analysis. The thread classification can help instructors in distinguishing resolved vs. unresolved discussions. Furthermore, state information helps instructors have insight on the process of discus-sion and facilitate them to understand the current state of the discussion. Such infor-mation supplies suggestions for instructors to decide when or whether to participate in the discussion.

Table 8. Precision, Recall and F-measure for thread classification

(a) Without state information

	Resolved			Unresolved		
	Precision	**Recall**	**F-value**	**Precision**	**Recall**	**F-value**
J48	0.92	0.94	**0.93**	0.71	0.66	**0.68**
SVM	0.87	0.98	0.92	0.81	0.39	0.52
LR	0.90	0.90	0.90	0.55	0.55	0.55

(b) With annotated state information

	Resolved			Unresolved		
	Precision	**Recall**	**F-value**	**Precision**	**Recall**	**F-value**
J48	0.95	0.99	**0.97**	0.94	0.75	**0.84**
SVM	0.90	0.98	0.94	0.85	0.50	0.63
LR	0.92	0.93	0.93	0.68	0.64	0.66

(c) With classified state information

	Resolved			**Unresolved**		
	Precision	**Recall**	**F-value**	**Precision**	**Recall**	**F-value**
J48	0.93	0.97	**0.95**	0.88	0.71	**0.78**
SVM	0.88	0.97	0.93	0.84	0.51	0.64
LR	0.91	0.90	0.90	0.64	0.66	0.65

5. Related work

There has been prior work on discussion analysis including use of speech act framework in modeling online discussions [3,4,5]. Some people focus on the roles that students play such as asking problems or answering other's questions [12,13]. Hidden Markov Model provides the framework for modeling the dialogue structure with hidden states [1,2,11]. They are closely related to our work, and we extend the existing framework by closely modeling the dialogue development and information exchange in Q&A discussions. In particular, we explicitly model problem and solution understanding phases as well as question and answer phases, and analyze the information exchange process using the state information. Graph-based approaches have been used in text mining, clustering and other related problems including labeling dialogue with tutors [1]. In order to facilitate the analysis of student discussions, we extend the existing work and represent a discussion thread as a graph model where each state in the model represents a message. There has also been work on machine classification of student online discussions [8,12,14] and results have been used to find meaningful dialogue patterns including features for critical thinking. Our work complements these results by closely examining and classifying Q&A processes.

6. Conclusion

We have presented a graph model for analyzing the discussion process and developed approaches for message state classification and thread characterization. The state information is used in analyzing frequent patterns and time intervals, and identifying different roles that instructors and students play in the Q&A process. Thread classifi-cation for resolved vs. unresolved problem is supported by the state information. As a next step, we plan to collect more data in order to obtain the more reliable classification result and explore additional improvement, including topic-based analysis of student problems. We plan to evaluate usefulness of the information with instructors.

References

[1] Boyer, K. E., Ha, E. Y., Wallis, M. D., Phillips, R., Vouk, M. A., & Lester, J. C. (2009). Discovering tutorial dialogue strategies with hidden Markov models. Proc. AIED 2009.

[2] Chi, M., VanLehn, K., & Litman, D. (2011). An Evaluation of Pedagogical Tutorial Tactics for a Natural Language Tutoring System: A Reinforcement Learning Approach. IJAIED.

[3] Forbes-Riley, K., & Litman, D. (2009). Adapting to student uncertainty improves tutoring dialogues. Proceedings of International Conference on Artificial Intelligence and Education.

[4] Fossati, D., Di Eugenio, B., Ohlsson, S., Brown, C., & Chen, L. (2010). Generating proactive feedback to help students stay on track. Proc. of ITS Conference 2010.

[5] Hirschberg, J. and Litman, D. (1993). Empirical Studies on the Disambiguation of Cue Phrases, Computational Linguistics, 19 (3).

[6] Kim, J., et al., (2006). Mining and assessing discussions on the web through speech act analysis. Workshop on Web Content Mining with Human Language Technologies at ISWC.

[7] McCallum, A. (2002). MALLET: a machine learning for language toolkit. http://mallet.cs.umass.edu

[8] McLaren, B. et al., Using Machine Learning Techniques to Analyze and Support Mediation of Student E-Discussions, Proceedings of AIED 2007.

[9] Sutton, C. (2006). An Introduction to Conditional Random Fields for Relational Learning.

[10] Witten, I.H., Frank, E.(2005). Data Mining: Practical machine learning tools and techniques. 2nd edn.

[11] Seo, S., Kang, J., Drummond, J. and Kim, J. (2011). Using Graphical Models to Classify Dialogue Transition in Online Q&A Discussions , Proceedings of AIED 2011.

[12] Mu, J., Stegmann, K., Mayfield, E., Rose, C., and Fischer, F. (2012). The ACODEA framework: Developing Segmentation and Classification Schemes for Fully Automatic Analysis of Online Discussions. Proc. CSCL 2012.

[13] Ravi, S. and Kim, J. (2007). Profiling Student Interactions in Threaded Discussions with Speech Act Classifiers. Proceeding of AIED 2007.

[14] Rus, V., Graesser, A., Moldovan, C., and Niraula, N. (2012). Automatic Discovery of Speech Act Categories in Educational Games, Proceedings of EDM.

[15] Saeys, Y., Inza, I., and Larranaga. P. (2007). A Review of Feature Selection Techniques in Bioinformatics. Bioinformatics, 23(19):2507-2517.

[16] Yoo, J. and Kim, J. (2012). Predicting Learners Project Performance with Dialogue Features in Online Q&A Discussions. Proceedings of ITS 2012.

Extending Collaborative Learning Modeling with Emotional Information

Olga C. Santos[1], Jesus G. Boticario[1], Raúl Cabestrero[2], Pilar Quirós[2]

[1] aDeNu Research Group. Artificial Intelligence Dept. Computer Science School. UNED
C/Juan del Rosal, 16. Madrid 28040. Spain
[2] Basic Psychology Dept. Faculty of Psychology. UNED.
C/Juan del Rosal, 10. Madrid 28040. Spain
[1]{ocsantos,jgb}@dia.uned.es, [2]{rcabestrero,pquiros}@psi.uned.es

Abstract. This work presents some initial ideas on a data mining based approach for building affective collaborative systems. In particular, we focused on the modeling issues involved in providing open affective student interaction models by using data mining techniques. The approach facilitates transferability and analysis without human intervention, and extends with emotional information previous data mining based developments.

Keywords: Collaboration, Data mining, Open models, Affective Computing.

1 Introduction

Given that affective issues play a significant role in e-learning scenarios [1, 2], in the context of the MAMIPEC project we are investigating emotions modeling in Computer Supported Collaborative Learning (CSCL), where either positive or negative emotions can emerge [3]. Positive ones are expected to bring about an increase in the number of users' interactions and accordingly the development of new collective generated knowledge. On the other hand, when individual aims collide with collective ones, negative emotions frequently arise. Under CSCL learners usually cope with more striking challenges than those present under face-to-face learning [4]. For instance, objectives of some group members can interfere with ones of others. Also, diversity in terms of levels of involvement, working styles and interaction modes frequently become overlapped within the group members. Additionally, the lack of previous common background and generally accepted point of view usually obstructs the way of getting cooperative solutions [3].

In this context, provided that Data Mining (DM) can be used for emotional information detection in CSCL [5], our goal is to extend the Collaborative Logical Framework (CLF) collaboration model [6] with emotional indicators and personality traits following a DM approach used in previous collaboration experiences [7].

2 Affective Collaborative Modeling approach

Personality traits and emotions play a key role in social and collaborative scenarios [4]. In this sense, it can be stated that personality can modulate the way the student participates on a given situation. For instance, some studies have found that participants that exhibit lower scores on extraversion and higher on mental openness prefer on-line learning tend [8]. Thus, in order to enrich adaptation in collaborative learning scenarios with affective support, the model has to take into account the user personality traits that can be influencing the user interaction behavior. It has also to consider the user affective state (i.e. pride, shame, curiosity, frustration, etc.) generated within the undergoing activity itself and the whole CSCL interaction. For this, i) context, ii) process and iii) assessment are considered key issues to model collaboration [9, 7].

The ***collaboration context*** affects students' potential and their capacity to collaborate. Information comes from data related to both students and the environment, which should be relevant to students' teamwork skills [10]. This information can be collected in the collaborative learning experience from an initial questionnaire (e.g., personal, academic and work-related data, study preferences, and personality traits).

The ***collaboration process*** involves features such as activity, initiative or acknowledgment. Relevant information can be obtained by analyzing students' interactions in communication tools such as forums [11] because of the close relationship that exists between students' collaboration and interactions. In this sense, previously we proposed a statistical analysis of the interactions in forums to discover some features that make students suitable for collaboration [6], namely student initiative, activity and regularity, as well as perceived reputation by their peers. Students' regularity indicators involve time variables because the interactions are considered over a period of time. In any case, these metrics are general in as much as they are based on non-semantic statistical indicators (e.g. number of replies, regularity of interventions, etc.) and thereby flexible enough to be potentially instantiated in diverse collaborative environments. In order to take into account affective information in these collaboration indicators, several information sources such as physiological data, keyboard and mouse interactions, explicit subjective affective information provided by learners, facial expression, etc. gathered while learners collaborate in the environment can be considered [12].

To cover aforementioned key issues, the approach we have been following offers ***collaborative assessment*** metrics based on DM process (clustering) to facilitate transferability and analysis without human intervention [7]. It also follows the open model strategy, which has shown its benefits in the educational context. This strategy uses scrutable tools that enable students to access inferred models and actively intervene in the modeling process [13], this way raising metacognitive information [7].

Our proposal for affective collaborative learning modeling is depicted in Fig. 1. In particular, to account for affective issues in the given *collaboration context* (user and environment), the approach has to be extended with an analysis of the affective reactions, elicited during the *collaboration process* within the ongoing collaboration *task* itself, and those due to the *interaction with peers* that feed the *collaboration assessment* and produce not only the *statistical indicators* proposed in [6] but also the add *affective ones*. The *affective indicators* are to be calculated with DM techniques in the light of the *collaboration assessment* by means of the *interaction content*: positive (proposing or suggesting; supporting or agreeing), negative (opposing or disagreeing) or ambivalent (information giving; inquiring; answering or specifying) as rated by both the emitter and the receiver (*interaction ratings* using weather 'overt' –subjective reports– or 'covert' –physiological or behavioral recordings– sources of information) [14]. To cope with *interactions latency*, it has to be taken into account if interaction are produced within certain time window or never take place at all –e.g. unanswered message–. On top of that, the *roles* could elicit an additional emotional reaction or modulate existing ones. Two different types have to be considered: *scripted* and *naturally emerged*. First ones are externally assigned, as a consequence of the statistical interaction indicators (i.e. information gatherer, moderator in the CLF *task* [6], etc.). Second ones emerge naturally in any collaborative work situations (i.e. task or social leadership or other types of roles that emerge in learning situations).

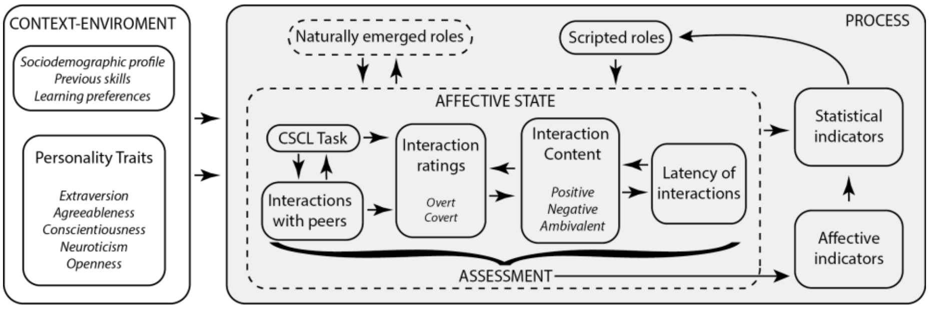


Fig. 1. Affective enriched statistical indicators in the open affective learner model

3 On-going work

To investigate how to enrich the statistical indicators with the affective ones, a CLF collaborative task was set out in Madrid's Week of Science 2012 with a total participation of 17 participants (including pilot experiments). They were asked to collaboratively solve one conundrum on a given time frame following three consecutive stages (*individual*: each participant proposes solution; *collaboration*: discussions and ratings among participants to enrich individual solutions; and *agreement*: solution proposed by moderator and discussed and rated by the rest of participants) while their collaboration interactions and affective information (i.e. personality questionnaires, physiological and behavioral recordings and subjective reports) are processed [6].

All these sources of information, along with the statistical indicators, deserve future analyses in order to refine and calibrate affective indicators and to articulate them using a DM approach. By introducing aforementioned affective issues the approach is expected to improve collaborative learning. In particular, based on our experience in developing educational recommender systems [15] those affective indicators detected will serve to develop affective educational recommendations.

Acknowledgments

Authors would like to thank the Spanish Ministry of Economy and Competence for funding the MAMIPEC project (TIN2011-29221-C03-01).

References

1. Kreijns, K., Kirschner, P.A., Jochems, W.: Identifying the pitfalls for social interaction in computer-supported collaborative learning environments: a review of the research. Computers in human behavior 19, 335-353 (2003)
2. O'Regan, K.: Emotion and e-learning. Journal of Asynchronous learning networks 7, 78-92 (2003)
3. Järvenoja, H., Järvelä, S.: Emotion control in collaborative learning situations: Do students regulate emotions evoked by social challenges? British Journal of Educational Psychology 79, 463-481 (2009)
4. Solimeno, A., Mebane, M.E., Tomai, M., Francescato, D.: The Influence of Students and Teachers Characteristics on the Efficacy of Face-to-Face and Computer Supported Collaborative Learning. Computers & Education 51, 109-128 (2008)
5. Calvo, R.A.: Incorporating Affect into Educational Design Patterns and Frameworks. Ninth IEEE International Conference on Advanced Learning Technologies, 2009 (ICALT 2009), 377-381 (2009)
6. Santos, O.C., Rodríguez, A., Gaudioso, E., Boticario, J.G.: Helping the tutor to manage a collaborative task in a web-based learning environment. In: AIED2003 Supplementary Proceedings, 153-162. (2003)
7. Anaya, A.R., Boticario, J.G.: Content-free collaborative learning modeling using data mining. User Modeling and User-Adapted Interaction 21, 181-216 (2011)
8. Santo, S.A.: Virtual learning, personality, and learning styles. vol. 62. ProQuest Information & Learning, US (2001)
9. Topping, K.J.: Methodological quandaries in studying process and outcomes in peer assessment. Learning and Instruction 20, 339-343 (2010)
10. van Gennip, N.A., Segers, M.S., Tillema, H.H.: Peer assessment as a collaborative learning activity: The role of interpersonal variables and conceptions. Learning and Instruction 20, 280-290 (2010)
11. Perera, D., Kay, J., Yacef, K., Koprinska, I.: Mining learners' traces from an online collaboration tool. Workshop Educational Data Mining, 13th International Conference of Artificial Intelligence in Education, 60–69 (2007)
12. Santos, O.C., Salmeron-Majadas, S., Boticario, J.G.: Emotions detection from math exercises by combining several data sources. LNAI, 7926, 742–745 (2013)
13. Bull, S., Gardner, P., Ahmad, N., Ting, J., Clarke, B.: Use and trust of simple independent open learner models to support learning within and across courses. In: Houben, G.-J., McCalla, G., Pianesi, F., Zancanari, M. (eds.) User Modeling, Adaptation, and Personalization, pp. 42-53. Springer (2009)
14. Nummenmaa, M.: Emotions in a web-based learning environment. Annales Universitatis Turkuensis, B 304 (2007)
15. Santos, O.C., Boticario, J.G., Manjarrés-Riesco, A.: An approach for an Affective Educational Recommendation Model. Recommender Systems for Technology Enhanced Learning: Research Trends & Applications. Manouselis, N., Drachsler, H., Verbert, K., Santos, O.C. (eds.). Springer, 2013 (in press).

Analysis of Emotion and Engagement in a STEM Alternate Reality Game

Yu-Han Chang, Rajiv Maheswaran, Jihie Kim, and Linwei Zhu

University of Southern California
Information Sciences Institute
Marina del Rey, CA 90292
{ychang,maheswar,jihie}@isi.edu, vic90228@gmail.com

Abstract. Alternate reality games (ARGs) are a promising new approach for increasing student engagement; however, automated methods for analyzing and optimizing game play are non-existent. We captured the player communication generated by a recent STEM-focused ARG that we piloted in a Los Angeles charter high school. We used shallow sentiment analysis to gauge the levels of various emotions experienced by the players during the course of the game. Pre/post-game surveys gauged whether the game narratives had any effect on student engagement and interest in STEM topics.

1 Introduction

Alternate Reality Games (ARGs) are a relatively new genre that has shown promise for engaging students in STEM learning activities. These transmedia experiences typically draw participants into fictional narratives, where players interact via various forms of social and traditional media, and frequently become part of the storyline themselves. They differ from traditional virtual reality computer games, where the entire story takes place in a fictional online world. In ARGs, the game world overlaps with the real world. Players visit real places, research the real world wide web, communicate with other players and fictional characters using real social media, phone, text messaging, and occasionally live encounters in the real world. For education, this novel game format has the potential to literally bring science activities and learning into the normal lives of students, emphasizing STEM relevance to the students context, surroundings, and community. The ARG brings the game space into the physical daily reality of students [?,?].

In this paper, we describe a pilot ARG we designed and implemented at USC Hybrid High in Fall 2012. We describe the ways in which we were able to capture player data, both by observing the players in game, and by validating these observations through pre and post game tests. In order for ARGs to truly support educational objectives, we need to be able to unobtrusively measure and understand the performance of players within the game, using only their in-game, visible interactions, such as website visitation and forum postings. Individual player assessment enables puppetmasters to tweak the game play to maximize engagement and educational outcome for each learner. Clearly AI and other computational techniques are needed to reach this goal, and this short paper only presents a summary of a small step in this direction.

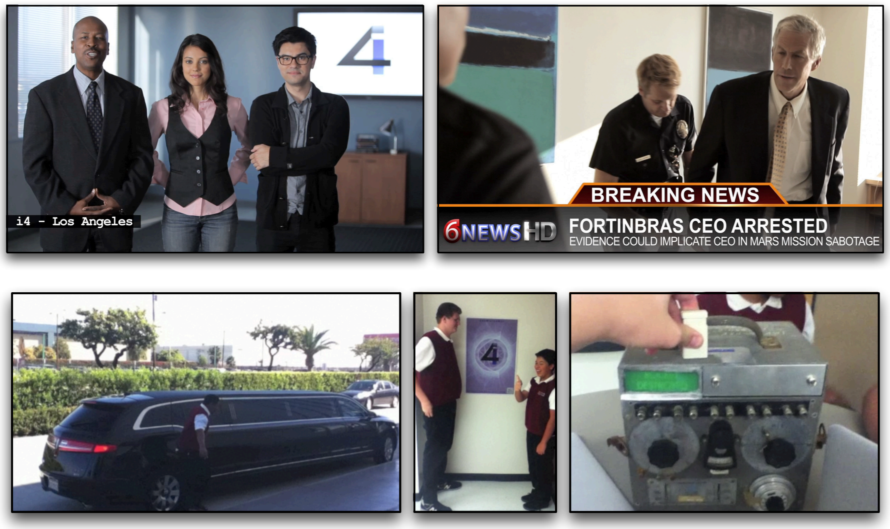


Fig. 1. (Top Left) The main characters in the game: William, Isa, and Rudy, (T. Right) The final story element in the game, where Fortinbras' CEO is arrested. (Bottom Left) Special trip to Space X facilities, (B. Center) Mysterious poster at USC Hybrid High, (B. Right) Device used to thwart Fortinbras.

USC Hybrid High ARG Pilot: Operation Daylight. In Fall 2012, we fielded a pilot alternate reality game, "Operation Daylight," at USC Hybrid High, a new charter high school with approximately 100 ninth graders in its inaugural class. The population is almost entirely minority and receive free/reduced lunches. The game focuses on π, an organization set up centuries ago to defend science. Its most recent incarnation, i4, needs students from USC Hybrid High to be their next generation, and the game begins with i4 recruiting and training students from the school. In the process, the students complete STEM-related activities to advance up the i4 recruitment ladder.

Gradually, the students uncover an evil plot by Fortinbras Industries that threatens their protagonist recruitment agents, the fictional characters Rudy Vanzant and Isa Figueroa, played by local actors in a variety of video sequences. This requires the students to put their newly learned skills to real use in order to save their friends Rudy and Isa. Figure **??** shows some of the elements used in the game. The game ran for approximately five weeks at USC Hybrid High, from 10/18/12 to 11/21/12. It was a completely optional activity that students could engage in if they chose to, with both online, at-school, and out-of-school elements. Students drove over 27,670 page views to the i4 website and posted 1394 messages to the i4 forum.

2 Methodology and Results

We used well-established scales for measuring student interest in STEM topics developed by OECD's Programme for International Student Assessment (PISA) [?]. Pre and post game surveys were developed using these scales, and administered to students at USC Hybrid High one week before the game commenced and one week after the game concluded. The surveys included approximately thirty questions where students would respond "Strongly Agree", "Agree", "Disagree", or "Strongly Disagree." The survey also included questions that established basic demographic information, as well as self-reported aspects of game play. In addition to the survey data, we also collected in-game data such as forum visits, messages posted, videos and pictures posted. We also used the Linguistic Inquiry and Word Count (LIWC) text analysis tool to process the messages [?] and detect whether they expressed a positive or negative sentiment, or whether the message contained anxiety, fear, or happiness.

Fifty-nine out of the 94 survey respondents indicated that they had heard of i4 and the Operation Daylight game. Twenty-three of the 29 students who signed up on the Operation Daylight website filled out surveys. Among students who played the game, they overwhelmingly thought the game increased their interest in science (48%) or did not change their already positive interest in science (47%). No one ended up having less interest in science.

These responses are corroborated with the students' answers to the OECD science interest questions. Figure **??** shows how the students' science interest levels changed from the beginning of the game to the end of the game, conditioned how often they visited the i4 forum, and on the average length of their posts on the forum. In these graphs, 0 corresponds to "Strongly disagree" (dislike science), and 3 corresponds to "Strongly agree" (like science). We see that there is a correlation between more visits and higher science interest level, as well as between longer posts and higher interest levels. There also appears to be a correlation between longer posts and a larger amount of increase in science interest.

Figure **??** shows that there is a correlation between forum activity and the major game events, such as the main characters being abducted. This suggests that these ARG story elements might promote the higher science interest levels described above. We also analyze the number of messages that contained certain percentages of message words that indicate positive or negative attitude, anxiety, fear, or sadness. It turned out that there is no clear pattern between the story elements and the production of particular categories of words, contrary to our expectation. For example, the abduction of the main character did not obviously produce more messages of fear or negativity. Generally the proportional levels of positive words stays constant during the game, and the levels of negative words

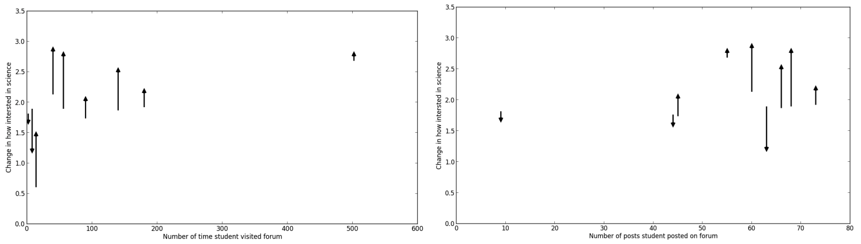


Fig. 2. (Left) Number of visits vs. change in science interest levels, (Right) Average length of forum postings vs. change in science interest levels.

stays quite low. The proportions of messages with varying levels positive words are also shown in Figure **??**. Due to lack of space here, a longer version of this paper will be posted at our website, http://cb.isi.edu.

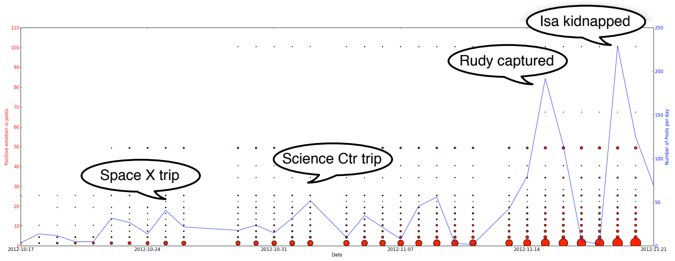


Fig. 3. Time showing the level of forum activity over the course of the game. The thin blue line denotes the number of posts in the forum on each day, the red circles denote how many of those messages contained a particular fraction of positive words.

Acknowledgements. This work was supported by NSF Award #:1224088.

References

1. E. Klopfer. *Augmented learning: Research and design of mobile educational games.* MIT Press, 2008.
2. Herbert W. Marsh, Kit-Tai Hau, Artelt Cordula, Baumert Jurgen, and Jules L. Peschar. Oecd's brief self-report measure of educational psychology's most useful affective constructs: Cross-cultural psychometric comparisons across 25 countries. *International Journal of Testing*, 6(4):311 360, 2006.
3. A. Moseley, N. Whitton, J. Culver, and K. Piatt. Motivation in alternate reality gaming environments and implications for learning. In *3rd European Conference on Games Based Learning.* Academic Conferences Limited, 2009.
4. J.W. Pennebaker. Putting stress into words: Health, linguistic, and therapeutic implications. *Behaviour research and therapy*, 31(6):539–548, 1993.

A Dashboard for Visualizing Deliberative Dialogue in Online Learning[1]

Tom Murray, Leah Wing[1], Beverly Park Woolf

School of Computer Science, [1]Legal Studies Dept.
University of Massachusetts, Amherst, MA

Contact: tmurray@cs.umass.edu

Abstract: New and emerging online trends in group education, work and communication have led to a dramatic increase in the quantity of information and connectivity without always supporting—and sometimes sacrificing—quality. An important opportunity is that online systems can include tools that directly support participants in having higher quality and more skillful engagements. We are evaluating dialogue software features that support participants directly and "dashboard" tools that support third parties (mediators, teachers, facilitators, moderators, etc.) in supporting higher quality deliberation. We will focus on our work in educational settings (college classes) and on our development of a Facilitators Dashboard that visualizes dialogue quality indicators for use as facilitation tools or participant social awareness tools. The Dashboard makes use of text analysis methods to highlight indicators of dialogue quality. We are particularly interested in supporting the "social deliberative skills" that interlocutors need to build mutual understanding and mutual regard in complex or contentious situations.

Keywords: Educational and Knowledge Building dialogue; deliberative skills; scaffolding; multiple representations; dashboards.

1. Introduction

New and emerging online trends in group education, work and communication have led to a dramatic increases in the quantity of information and connectivity without always supporting—and sometimes sacrificing—their quality. An important opportunity is that online systems can include tools that directly support participants in having higher quality and more skillful engagements. We are building and evaluating dialogue software features that support *participants* directly and "dashboard" tools (Few, 2007) that support *third parties* (mediators, teachers, facilitators, moderators, etc.) in supporting higher quality deliberation among participants. In this paper we will focus on our work in educational settings (college classes) and on our development of a Facilitators Dashboard that visualizes dialogue quality indicators for use by

[1] A longer version of this short paper appears at www.socialdeliberativeskills.com/papers.

either third parties or participants. We are particularly interested in supporting the "social deliberative skills" that interlocutors need to build mutual understanding and mutual regard in complex or contentious situations (Murray et al., 2013A, B). Prior attempts to facilitate leaner dialogue using visualization and analysis tools, e.g. Asterhan & Swhwatz (2010) and De Groot et al. (2007), tend to focus on argumentation skills, and our work extends or complements this work by focusing on skills more related to mutual understanding and cognitive empathy. Communication, collaboration, and knowledge building have many facets; and we focus our research on a specific area: supporting the social deliberative skills and behaviors that allow interlocutors to build mutual understanding (or "negotiate meaning") in complex or contentious contexts. Recent advances in computational psycholinguistics allow for a more systematic and deeper analysis of dialogues, that is necessary to uncover subtle cues that might be diagnostic of critical deliberation characteristics. In Xu et al. (2013) we report on our work in developing computational methods to measure deliberative skills from online discussions, which have shown promising results. In this paper we will describe our progress and plans for displaying the results of such text analysis in the Dashboard.

2. Dashboard Diagram Pane: Visualizing Key Indicators

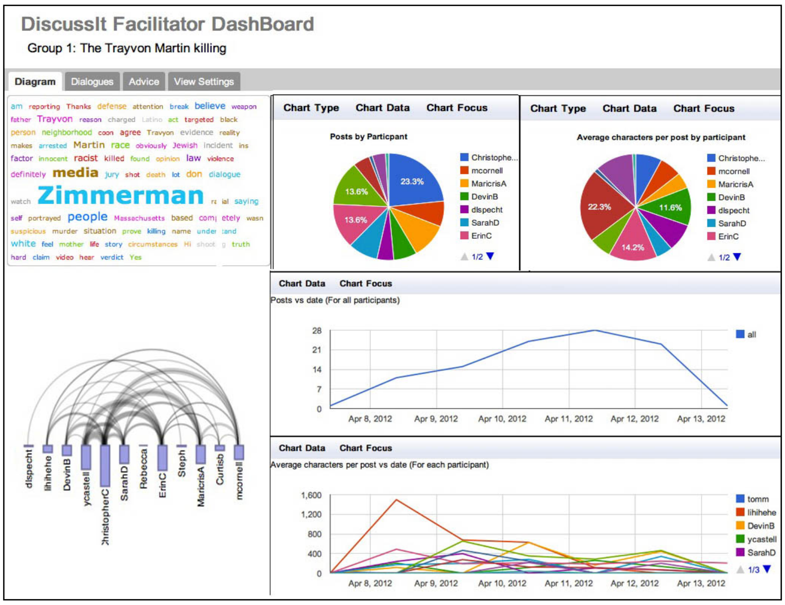

DiscussIt Facilitator DashBoard
Group 1: The Trayvon Martin killing
Diagram
Dialogues
Advice
View Settings
am reporting Thanks defense attention break believe weapon father Trayvon reason charged Latino act targeted black person neighborhood coon agree Travyon evidence reality makes arrested Martin race obviously Jewish incident ins factor innocent racist killed found opinion law violence definitely media jury shot death lot don dialogue watch Zimmerman ra ial saying self portrayed people Massachusetts based com etely wasn suspicious murder situation prove killing name under tand white feel mother life story circumstances Hi shoot g truth hard claim video hear verdict Yes
dispecht
lihihehe
DevinB
ycastell
ChristopherC
SarahD
Rebecca
ErinC
Steph
MaricrisA
Curtisb
mcornell
Chart Type
Chart Data
Chart Focus
Posts by Particpant
23.3%
13.6%
13.6%
Christophe...
mcornell
MaricrisA
DevinB
dispecht
SarahD
ErinC
1/2
Chart Type
Chart Data
Chart Focus
Average characters per post by participant
22.3%
11.6%
14.2%
Christophe...
mcornell
MaricrisA
DevinB
dispecht
SarahD
ErinC
1/2
Chart Data
Chart Focus
Posts vs date (For all participants)
28
21
14
7
0
Apr 8, 2012
Apr 9, 2012
Apr 10, 2012
Apr 11, 2012
Apr 12, 2012
Apr 13, 2012
all
Chart Data
Chart Focus
Average characters per post vs date (For each participant)
1,600
1,200
800
400
0
Apr 8, 2012
Apr 9, 2012
Apr 10, 2012
Apr 11, 2012
Apr 12, 2012
Apr 13, 2012
tomm
lihihehe
DevinB
ycastell
SarahD
1/3

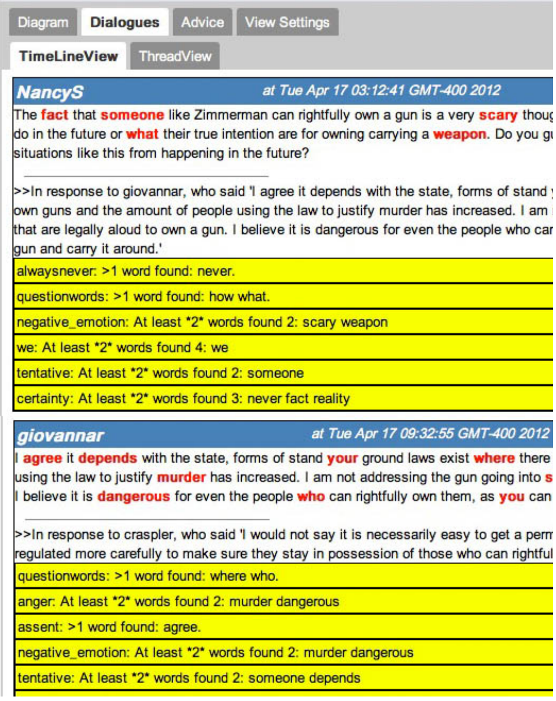
Figure 1: Facilitator Dashboard: Diagram Pane

We have prototyped a Facilitators Dashboard that provides parties a "bird's-eye view" of the state and flow of online engagements. See Figure 1 which shows tools in the "Diagram" tab of the Dashboard. Similar to Iandoli et al., De Groot et al., we visualize user, interaction, and content information, including participation levels, reply networks, and content or theme overviews—in both static and trend (timeline) visualizations. At a more ambitious level, we also use text analysis to identify skillful (or nonskillful) deliberation, emotional tone or sentiment. Further, we have made early forays into automatically identifying dialogue phases (e.g. introductions, deliberation, impasses, persuasion) and turning/infection points or opportunities for intervention (e.g. silences or non-responsiveness, changes of phase or tone, sudden emotional tensions in multiple participants) (Xu et al. 2013).

Figure 2 shows data from a classroom discussion about the fatal shooting of Trayvon Martin by George Zimmerman which was a hot topic in the news during the time of this activity. When the facilitator begins using the Dashboard they select from a list of the deliberation projects, classes, or discussion groups registered with the Mediem software and the Dashboard (not shown in the Figure). Pie and bar charts show participation levels (number of participant posts and average size of posts). Timelines show trends in these same metrics. A social network diagram shows who is replying to whom, with the thickness of the lines proportional to the number of replies. A "word cloud" graphically shows word frequencies through font sizes (the color and location of the words has no meaning in this representation).

3. Dialogue and Advice Panes: Text Analysis

As mentioned above, one component of our project is researching automatic text analysis and machine learning algorithms (and soon also relationship networks) to identify deliberative skill, other indicators related to dialogue quality, and trends or opportunity points (and see Rosé et al. 2008). Text analyses methods have advanced significantly in recent years. According to Graesser et al. (2009) the "increased use of automated text analysis tools can be attributed to landmark advances in such fields as computational linguistics, discourse processes… , cognitive science…, and corpus linguistics…" (p. 34). We are using three types of technologies. The first two, LIWC (Pennebaker et al, 2007) and Cohmetrix (Graesser et al., 2009), are pre-existing text analysis tools that take text segments as inputs and output dozens of measurement or classification metrics. The third technology is a set of machine learning methods we are using that take text, reply and demographic information, and some of the LIWC and Cohmetrix out-



NancyS at Tue Apr 17 03:12:41 GMT-400 2012

The fact that someone like Zimmerman can rightfully own a gun is a very scary thoug
do in the future or what their true intention are for owning carrying a weapon. Do you gu
situations like this from happening in the future?

>>In response to giovannar, who said 'I agree it depends with the state, forms of stand y
own guns and the amount of people using the law to justify murder has increased. I am
that are legally aloud to own a gun. I believe it is dangerous for even the people who car
gun and carry it around.'

alwaysnever: >1 word found: never.

questionwords: >1 word found: how what.

negative_emotion: At least *2* words found 2: scary weapon

we: At least *2* words found 4: we

tentative: At least *2* words found 2: someone

certainty: At least *2* words found 3: never fact reality

giovannar at Tue Apr 17 09:32:55 GMT-400 2012

I agree it depends with the state, forms of stand your ground laws exist where there
using the law to justify murder has increased. I am not addressing the gun going into s
I believe it is dangerous for even the people who can rightfully own them, as you can

>>In response to craspler, who said 'I would not say it is necessarily easy to get a perm
regulated more carefully to make sure they stay in possession of those who can rightful

questionwords: >1 word found: where who.

anger: At least *2* words found 2: murder dangerous

assent: >1 word found: agree.

negative_emotion: At least *2* words found 2: murder dangerous

tentative: At least *2* words found 2: someone depends

Figure 2: Dashboard: Dialogue Pane

puts as input or training features, and output classification analysis (e.g. whether a segment of text demonstrates good "deliberative skill" or "self reflection").

4. Conclusions

We have described a novel Facilitators Dashboard tool that visualizes dialogue quality indicators for use as facilitation tools or participant social awareness tools that includes textual analysis and described our initial attempts to use it in educational settings. We are particularly interested in supporting the "social deliberative skills" that interlocutors need to build mutual understanding and mutual regard in complex or contentious situations. Developing methods to scaffold SD-skills in online deliberation, for participants and third parties, could have an impact in many online contexts; e.g. knowledge-building, situated learning, civic engagement, and dispute resolution. Students engaged in extended collaborative knowledge building, discussion, or problem solving eventually encounter moments of tension in which they are challenged to understand each other's perspectives and opinions. Engaging with others on complex topics requires not only learning the relevant facts and concepts and making logical inferences but also, engaging with the perspectives and opinions of others who may not share one's views or goals. Doing so requires skills that can be systematically supported. Our work points to how such skills can be supported in online deliberation, collaboration, and dispute resolution—in educational settings and beyond.

5. References

Asterhan, C. S., & Schwarz, B. B. (2010). Assisting the facilitator: Striking a balance between intelligent and human support of computer-mediated discussions. In Workshop on opportunities for intelligent and adaptive behavior in collaborative learning systems (p. 1).

De Groot, R., et al.. (2007). Computer supported moderation of e-discussions: the ARGUNAUT approach. In Proceedings of the 8th international conference on Computer supported collaborative learning (pp. 168-170), July, 2007. International Society of the Learning Sciences.

Few, S. (2007). “Dashboard Confusion Revised.” Perceptual Edge, March 2007. From http://www.perceptualedge.com/articles/03-22-07.pdf.

Graesser, A.C., McNamara, D.S., & Louwerse, M.M. (2009). Methods of Automated Text Analysis. Chapter 4 in the Handbook of Reading Research, Volume IV, edited by Michael L. Kamil, P. David Pearson, Elizabeth B. Moje, and Peter Afflerbach. Routledge.

Murray, T., Stephens, A.L., Woolf, B.P., Wing, L., Xu, X., & Shrikant, N. (2013a). Supporting Social Deliberative Skills Online: the Effects of Reflective Scaffolding Tools. Proceedings of HCI International 2013, July, 2013, Las Vegas.

Murray, T., Xu, X. & Woolf, P.B. (2013b). An Exploration of Text Analysis Methods to Identify Social Deliberative Skills. In Proceedings of AIED-13, July, 2013, Memphis, TN.

Pennebaker, J. W., Chung, C. K., Ireland, M., Gonzales, A. L., & Booth, R. J. (2007). The development and psychometric properties of LIWC2007. Austin, TX: www.LIWC.net.

Rosé, C., Wang, Y. C., Cui, Y., Arguello, J., Stegmann, K., Weinberger, A., & Fischer, F. (2008). Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning. International journal of computer-supported collaborative learning, 3(3), 237-271.

Xu, X., Murray, T., Smith, D. & Woolf, B.P. (2013) . Mining Social Deliberation in Online Communication: If You Were Me and I Were You . Proceedings of EDM-13, Educational Data Mining, July, 2013, Memphis, TN.

Designing OLMs for Reflection about Group Brainstorming at Interactive Tabletops

Andrew Clayphan, Roberto Martinez-Maldonado, and Judy Kay

School of Information Technologies, The University of Sydney, NSW, 2006, Australia
{andrew.clayphan,judy.kay}@sydney.edu.au
roberto@it.usyd.edu.au

Abstract. Brainstorming is a valuable and widely-used group technique to enhance creativity. Interactive tabletops have the potential to support brainstorming and, by exploiting learners' trace data, they can provide Open Learner Models (OLMs) to support reflection on a brainstorming session. We describe our design of such OLMs to enable an individual to answer core questions: C1) how much did I contribute? C2) at what times was the group or an individual stuck? and C3) where did group members seem to 'spark' off each other? We conducted 24 brainstorming sessions and analysed them to create core brainstorming models underlying the OLMs. We evaluated the OLMs in a think-aloud study designed to see whether learners could interpret the OLMs to answer the core questions. Results indicate the OLMs were effective and that it is valuable, that learners benefit from guidance in their reflection and from drawing on an example of an excellent group's OLM. Our contributions are: i) the first OLMs supporting reflection on brainstorming; ii) models of brainstorming that underlie the OLMs; and iii) a user study demonstrating that learners can use the OLMs to answer the core reflection questions.

Keywords: Open Learner Models, Brainstorming, Reflection

1 Introduction

Brainstorming is a valuable and widely used technique to produce creative solutions to a problem [11]. It is particularly useful when innovation is needed to break out of established ways of thinking, to generate new ideas. When the brainstorming activity is run in small groups, it encourages participants to contribute to the free flow of ideas around a topic, bringing their own creativity, experiences or expertise into play, and increasing the opportunities of enhanced production of rich ideas for the solution. Osborn, the creator [16] promoted the use of brainstorming for creativity. He emphasised that, to be effective, core rules should be followed to reduce members social inhibitions and stimulate idea generation: the focus should be on the *quantity* of ideas; there should be *no early evaluation*; particularly *no criticism*; and *un-usual or divergent ideas welcomed.* Therefore, all participants are encouraged to contribute fully and equally. Discussion should be limited to cases where people are *stuck* and cannot create ideas.

Multi-touch interactive tabletops have proved effective in facilitating face-to-face brainstorming in small-groups [6]. They can support free flow of ideas by providing a shared group interface so that people can generate many ideas in parallel, then interact with digital representations of these ideas, and save the generated ideas offering all team members equal opportunities to contribute [7]. A less explored potential of interactive tabletops is to exploit data about the interaction to capture the processes through the brainstorming session and then show key information about group and individual performance as Open Learner Models (OLMs) [4]. OLMs are those representations of learners' (knowledge, developed skills, performance, understanding, etc.) that are accessible to the learner or group of learners they represent. They can then serve several roles, including support for reflection [5], formative assessment [2] and facilitate collaborative interaction [3]. We particularly focus on the potential value of Open Learner Models (OLMs) as a driver for individuals to reflect on their individual and group performance after a brainstorming session.

The rest of the paper is organised as follows. Next, we outline related research work on OLMs for group work and interactive tabletops. Section 3 describes ScriptStorm, our tabletop system for brainstorming. Section 4 describes the design of our OLM and our evaluation is presented in Section 5. We conclude with a discussion of the results and future work.

2 Related Work

OLMs have been used to facilitate group interaction by enabling learners to identify peers for collaboration [2]. It has been shown that there is value in providing multiple OLM representations helping support higher levels of reflection, because different learners prefer different forms of OLMs, particularly to meet differing concerns [12]. There has also been some exploration of how an ITS can help a learner in brainstorming [18]. Some of the ways such systems can be beneficial is to help learners realise whether they followed recommended practices for brainstorming effectively, particularly in terms of avoiding early evaluation and whether group members suffered blocks [9] in the session.

Some research has started to explore OLM visualisations that represent collaborative learning at interactive tabletops. Martinez-Maldonado et al. [14] validated a set of such OLMs with teachers, showing they could identify the level of collaboration. Al-Qaraghuli et al. [1] presented a visualisation that showed detailed information of students actions at a tabletop over time to foster deep analysis of the process they followed. These authors also provided a small pie chart on the interactive tabletop showing students a real time indication of each learners' participation. Martinez-Maldonado et al. [15] built a dashboard OLM for the teacher to see real-time information about aspects of collaboration for multiple groups in a classroom of interactive tabletops. These examples aimed either to show 'learner models' to the teacher or have been used for research purposes only. Our work goes beyond this by evaluating OLMs that can be presented to learners at an interactive tabletop to promote self-reflection at the end of a brainstorming session. In this sense, it is similar to Do-Lenh's [10] work, for a multi-tabletop classroom where a simple form OLM gave indication of the progress of each group on a wall display for all students to see.

3 Foundations for design of the Open Learner Models

The need for OLMs to support reflection at a tabletop for brainstorming was identified when we evaluated *Scriptstorm* [8], a scripted tabletop brainstorming system (Figure 1). ScriptStorm had three main stages: (1) idea generation – the "storming" to create ideas; (2) idea categorisation – to organise ideas under category headings; and (3) reflection – to support learners by reflection-on-action [17]. While the scripting proved valuable, the reflection stage did not enable participants to appreciate how well they had followed the recommended brainstorming process. We analysed the data from the study to explore how to create OLMs that could provide more effective support for reflection.

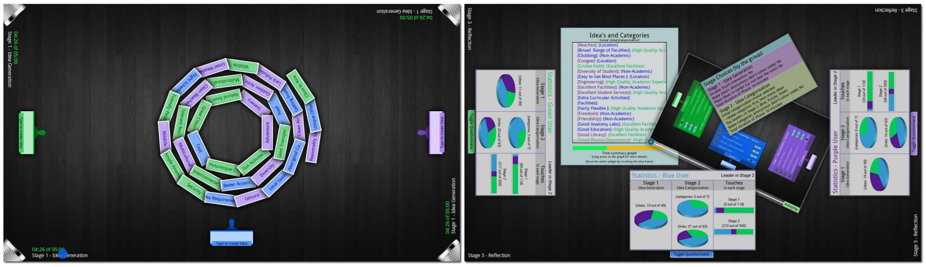


Fig. 1. ScriptStorm: Idea Generation Stage (left), Reflection Stage (right).

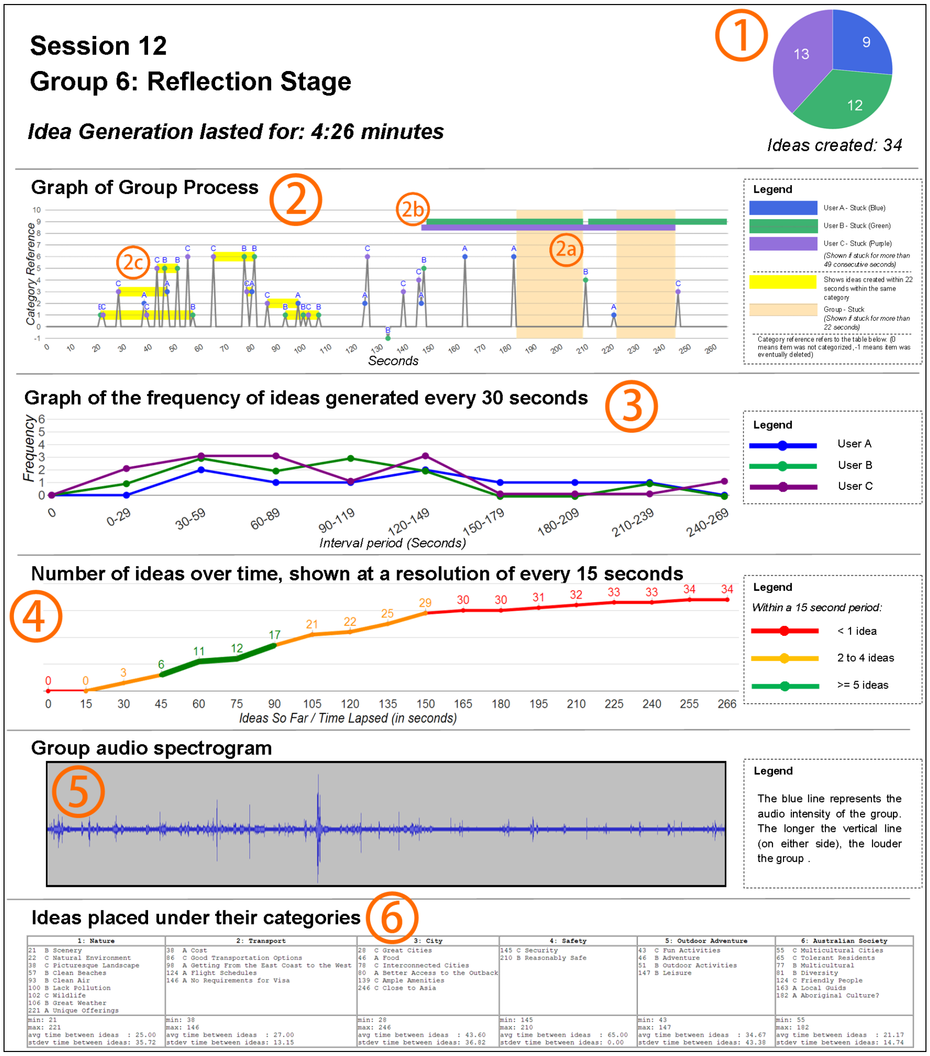
We describe *Scriptstorm*, the study, the data collected and the analyses conducted for this work. *Scriptstorm* uses physical keyboards at a multi-touch tabletop. Figure 1-left shows an example table-shot after a group has created several ideas, visible in a circle at the centre of the table. This layout reduces the sense of ownership of ideas and the circular orientation avoids favouring any one user's reading. Ideas are colour coded to indicate the author, giving an indication of each person's level of contribution. Figure 1-right shows the elements available in the reflection stage. Each user has a set of charts showing each person's contributions. Pie charts show how many ideas each person made in Stage 1, how many categories and classification of ideas into them in Stage 2. A bar chart shows touches by each user in each stage. There is a list of the ideas with their categories in the middle, details of the scripting choices made and a replay of the table. Touches were logged by the tabletop and linked to the user making use of a depth camera [13].

The evaluation had 12 groups, each with 3 people (36 participants, 22 male, 14 female, all university students, from diverse degrees – medicine, social science and computer science, aged 19-30, mean age 23). Each group did 2 brainstorms, counter-balanced on scripting condition and topic. Each group was instructed of the rules of brainstorming to follow. Careful analysis of the data indicated the topic and scripting conditions were comparable, making for 24 sessions of data for analysis. All sessions were video recorded.

We analysed the study data to create a model of brainstorming as a foundation for the OLMs. This model provides a bound on the time-between-ideas when the brainstorm is running well. This is important since we can then use it to automatically determine when a group or individual is stuck, and determine if ideas from different users are sparked off each other. Groups created 16 to 104 ideas per session (average = 48; standard deviation = 24), average time between ideas 7.32 seconds (SD = 4.2) range of 2.88 − 17.93 seconds. We explored the frequency distribution of times, a single hump, slightly left of the peak at 7 seconds. For the individual, average time between ideas was 26.16 seconds (SD = 21.64), range 5.75 − 110.5 seconds. We arrived at a maximum idle time for a group before being classified as stuck as 22 seconds (mean group time difference + SD), and for an individual 49 seconds (three times the mean). We also used 22 seconds to scan for ideas that potentially sparked other ideas. These values are used as measures in our OLM to highlight interesting periods. Additionally we analysed output in terms of 15 second periods, resulting in a range of 0 to 13 ideas, accounting for outliers, the average being 4 ideas. We used this in our OLM as the basis for a colour coding scheme (red, orange, green), representing: below, average and above average performance.

4 Open Learner Model Design

We needed to enable learners to answer our core questions: *C1) how much did I contribute? C2) at what times was the group or an individual stuck?* and *C3) where did group members seem to 'spark' off each other?* To help learners find answers to these questions, we designed the OLMs in Figure 2 to present six different views of the user trace data. The pie chart (chart 1) shows the number of ideas each person created (C1). Following, there are four aligned timelines. Chart 2 shows when each idea was created with by a dot, the colour of which indicating authorship. The vertical axis indicates the category from the second phase of the brainstorm. Stuck periods are shown as coloured rectangles for the group (2a) and coloured bars for individuals (2b). In the figure the group got stuck twice between 183-209 and 222-244 seconds, the green user stuck between 148-209 and 211-266 seconds, the purple user stuck between 146-245 seconds and the blue user not stuck at all. To model where people sparked off each other, we identified cases where one persons idea was closely followed by another according to the category classification. This measure is shown with yellow bars (2c). There are seven of these in the diagram, for example on category reference 6 between 65-81 seconds (ideas 65s-C, 77s-B, 81s-B). This measure is clearly an inexact measure that is sensitive to the particular categories chosen, however it is indicative of sparking and showing it in an OLM helps users consider this aspect (C2,3). The next timeline (chart 3) shows the performance of each learner in 30 second snapshots (C1,2). The timeline after that (chart 4) shows cumulative progress with segments colour coded according to the rate of contribution (C2). The final timeline (chart 5) is a spectrogram indicating when a group was talking. Learners were instructed to call out each idea they generated in the idea generation stage and we expected discussion if a group was stuck (C2,3). The last view (label 6) is a table with categories and associated ideas annotated with author and time of creation.



1: Nature	2: Transport	3: City	4: Safety	5: Outdoor Adventure	6: Australian Society
21 B Scenery 22 C Natural Environment 38 C Picturesque Landscape 57 B Clean Beaches 93 B Clean Air 100 B Lack Pollution 102 C Wildlife 106 B Great Weather 221 A Unique Offerings	38 A Cost 86 C Good Transportation Options 98 A Getting From the East Coast to the West 124 A Flight Schedules 146 A No Requirements for Visa	28 C Great Cities 46 A Food 78 C Interconnected Cities 80 A Better Access to the Outback 139 C Ample Amenities 246 C Close to Asia	145 C Security 210 B Reasonably Safe	43 C Fun Activities 46 B Adventure 51 B Outdoor Activities 147 B Leisure	55 C Multicultural Cities 65 C Tolerant Residents 77 B Multicultural 81 B Diversity 124 C Friendly People 163 A Local Guids 182 A Aboriginal Culture?
min: 21 max: 221 avg time between ideas : 25.00 stdev time between ideas: 35.72	min: 38 max: 146 avg time between ideas : 27.00 stdev time between ideas: 13.15	min: 28 max: 246 avg time between ideas : 43.60 stdev time between ideas: 36.82	min: 145 max: 210 avg time between ideas : 65.00 stdev time between ideas: 0.00	min: 43 max: 147 avg time between ideas : 34.67 stdev time between ideas: 43.38	min: 55 max: 182 avg time between ideas : 21.17 stdev time between ideas: 14.74

Fig. 2. Open Learner Model Visualisations.

5 Evaluation

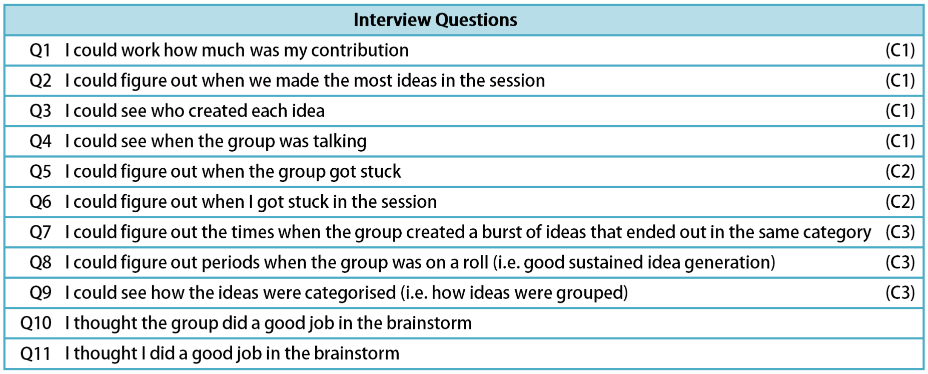
We conducted an interview/think-aloud study with 15 participants drawn from the earlier brainstorming study (10 male, 5 female, age range 21-30, mean age 24), each interviewed separately. The study consisted of analysing 3 anonymised brainstorming sessions from the earlier study (the same 3 anonymised sessions across all interviews). The visualisations were presented on laminated A3 sheets of paper to aid visibility, and contained the different OLMs like the one shown in Figure 2 – which allowed learners to quickly point to the different items when answering the questions. These questions, listed in Table 1, investigated whether participants, could obtain information, about individual/group contributions (Q1−4), if they could identify periods when the group or its members got 'stuck' (Q5-6) or if they could define whether the group members sparked off of each other (Q7−9). Questions 10 and 11 served as self-assessment of the group and individual performance respectively. The interview questions (Table 1) linked to our core research questions as shown in Table 2. The interview process had the following steps:

Step 1 Participants were asked to pretend to be a learner that produced 13 ideas in a group who made 34 ideas (i.e. to be the purple user in Figure 2), and answer the questions in Table 1.

Step 2 Participants were shown a numerically well performing group whom created 80 ideas and asked to review their answers to Q10 and Q11. We did this to see if people would change their response, given extra information.

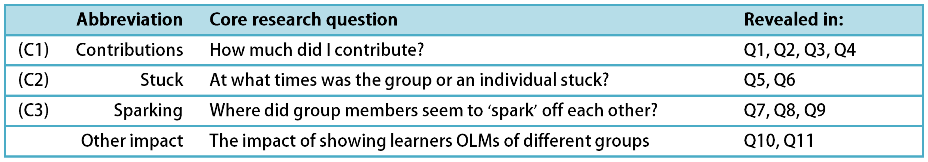
Step 3 Participants were asked to pretend to be a learner with 52 ideas in a group with 98 ideas, and answer the questions in Table 1.

Step 4 Participants were asked three general questions: (1) Whether they would like to see these visualisations as part of a reflection stage on a tabletop; (2) Whether they thought the visualisations would enable a group to become more effective; and (3) If you were a user with a low number of ideas, would the visualisations make you more aware and conscious about your performance.



	Interview Questions	
Q1	I could work how much was my contribution	(C1)
Q2	I could figure out when we made the most ideas in the session	(C1)
Q3	I could see who created each idea	(C1)
Q4	I could see when the group was talking	(C1)
Q5	I could figure out when the group got stuck	(C2)
Q6	I could figure out when I got stuck in the session	(C2)
Q7	I could figure out the times when the group created a burst of ideas that ended out in the same category	(C3)
Q8	I could figure out periods when the group was on a roll (i.e. good sustained idea generation)	(C3)
Q9	I could see how the ideas were categorised (i.e. how ideas were grouped)	(C3)
Q10	I thought the group did a good job in the brainstorm	
Q11	I thought I did a good job in the brainstorm	

Table 1. Interview questions investigating the usefulness of the group OLMs.



	Abbreviation	Core research question	Revealed in:
(C1)	Contributions	How much did I contribute?	Q1, Q2, Q3, Q4
(C2)	Stuck	At what times was the group or an individual stuck?	Q5, Q6
(C3)	Sparking	Where did group members seem to 'spark' off each other?	Q7, Q8, Q9
	Other impact	The impact of showing learners OLMs of different groups	Q10, Q11

Table 2. Relationship between research questions and interview questions.

Responses were given on a 6-point Likert scale (1 for strongly disagree, 6 for strongly agree). Participants were instructed to point to any items (the charts/table) that influenced their response as well as provide an explanation for each item chosen. Results are summarised in Table 3.

		Contributions				Stuck		Sparking			Other impact	
Questions		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
Step 1 **34 idea group**	Likert	5.07	5.53	4.87	5.40	5.67	5.87	4.20	5.00	5.20	4.40	4.73
	Item	1,3	4,3	2,6	5,2	2,4	2,3	2,6	4,2	6,2	1,2	3, 2
Step 2 **80 idea group**	Likert										**3.40**	**4.40**
Step 3 **98 idea group**	Likert	5.53	4.93	5.33	5.20	5.27	5.40	**5.20**	5.20	5.27	5.20	5.60
	Item	1,6	4,2	6,2	5	2,4	2,3	2,6	4,3	6,2	1,4	1,3

Table 3. Results of the interview. Item refers to those as labelled in Figure 2, briefly: 1–pie chart; 2–graph of group process; 3–graph of frequency of ideas; 4–number of ideas over time; 5–group audio spectrogram; and 6–the table. The two most commonly referenced items are included. Bold indicates a statistically significant change from Step 1 to 2 (Q10,11) and from Step 1 to 3 (Q1-9).

Most of the learners agreed that the OLM visualisations provided key information about the group brainstorm ($\geq$4.20 across the Likert scores). While participants thought aloud, more than half mentioned ease of understandability, especially by the time they saw the third groups OLMs. Some users had initial difficulties understanding certain visualisations, for example four users initially found chart 2 to be very complex, though by the end of the activity, only two of these four still found the visualisation complex.

6 Discussion

6.1 Group members contributions to the brainstorm

In the absence of a benchmark to compare the number of ideas generated, participants determined if a group did a good job, by judging levels of equality, referring to charts 1 and 3. When additional group OLMs were introduced, participants focused on the amount of ideas produced. For individual contribution – Q1, participants drew from charts 1 and 3 and the table. Chart 1 presented overall contribution in a simple form: *P4– "easy to understand"*; *P5– "very clear"*; and *P3– "I have the biggest cut of the pie"*. Chart 3 revealed contributions over time: *P6– "I generated the most ideas in the first 90 seconds"*; and *P2– "I compared the number of ideas generated and saw that I created just as many as the others"*. For determination of active periods (Q2), 12 people (P1,2,3,4,8,9,10,12,13,15) consulted chart 4 – referencing the colour scheme. A small number of participants referred to chart 3, looking at times when frequency of ideas generated was high across all members. For whom created each idea – Q3, chart 2 and the table were referenced. For chart 2 – the coloured dots representing authors were used (P1,5,7,8,9,10,11), and for chart 6 – the author written alongside the idea (P2,3,6,12,14). Overall, the following were referred to the most: chart 1 – for individual contribution; chart 2 – for whom created each idea; and chart 4 – for periods containing a large number of ideas.

6.2 Periods where the group or individuals got stuck

For Q5 – identify when the group was stuck and Q6 – identify when individuals were stuck, the average Likert score was above 5 (Q5: 5.70 & 5.27, Q6: 5.87 & 5.40). Participants utilised charts 2, 3 and 4. For chart 2 – the shaded regions and horizontal bars were referenced (P1,7,8,9,10,11,12,15): *P9– "I looked at the interval between ideas"*; *P3– "I looked for the shades to see if they were stuck, when I couldn't see any, so I checked this one [chart 4] to see if there were any red lines"*; and *P10– "easy to see when I was stuck, because of the highlights"*. For chart 3 – participants looked for when groups tapered off, shown as dips (P1,2,3,4,6,9,14): *P2– "The graph plateaued at the end, showing me they got stuck"*, similarly in chart 4 – the gradient of the line combined with the colour coded segments (P4,5,9,11,13): *P5– "because of the red"*. Overall, chart 2 proved to be most useful for identifying stuck periods. These observations reinforce the usefulness of the information added from our brainstorming model, in providing potentially useful visual indicators to learners. These indicators (the shading, bars and coloured segments) can be the basis for discussion, reflecting on actions that led to identified periods of inactivity.

6.3 Evidence that group members 'sparked' off of each other

Question 7 asked whether a burst of ideas ended up in the same category. For this question, chart 2 was referenced, but with mixed responses. 8 participants said the yellow highlight in chart 2 was obvious: *P13– "I looked at the yellow lines, as it easily caught my attention"*, but 4 participants did not find the highlight obvious and instead horizontally scanned the grey line present on each row. Three participants mentioned the table, and said that if they spent more time they could of worked out which ideas from whom sparked other ideas, but were off put by the presentation, being heavy in text, compared to the other items. Determining when a large number of ideas was created, without the constraint of them being in the same category, participants shifted focus to chart 4. Overall, chart 2 was most useful for showing when members sparked off of each other. This can be used as a starting point for discussion in a reflection stage to talk about sparking and what led to it, and how often it occurred.

6.4 The impact of showing learners OLMs of different groups

Participants were shown an example of a particularly productive group after the first group and asked to reflect on Q10 and Q11, questions which related to performance. For group performance (Q10), upon seeing another group, with a higher number of ideas, 8 people (P2,3,7,9,10,13,14) downgraded their answer with an average reduction of 2 Likert points, resulting in a statistically significant decrease (from 4.4 to 3.4), representing a switch from the agree to the disagree side of the Likert scale. The primary reason cited was the difference in the number of ideas created (P2,3,7,9,10), and the lack of stuck periods in the new group (P13,14). Three participants (P11,12,15) kept their original answer stating whether a group performed well is more complex than a numerical figure, raising issues of group dynamics, questions about quality, and requested other group OLMs to have more information to compare against: *P12– "I only have 2 groups to go off, not a complete average, also I don't know if their quality was the same"* and *P7– "The first group generated longer multiple word ideas, while this group created single word ideas, I think that's why the first group had less ideas"*. For Q11, 5 participants changed their response, with the bulk of participants pointing out that the user with 13 ideas (the purple user) made the most ideas of the group (P1,4,8,9,11,13,15); and *P9– "purple did a good job in his group, and his performance is also dependent on his team members, so I decide to keep my original answer the same"*. Two participants (P6,11) mentioned they wanted to have an average value, to put the number of created ideas into perspective.

These comparisons point to the fact that participants are not only influenced through their own contributions within a group, but also the performance of related groups brainstorming. An apparent strong feeling of success can be changed when exposed to other group OLMs. This is helpful in promoting reflection, in order to promote a deeper understanding of performance, and also possibly to inspire learners to develop skills to improve themselves.

Overall, the impact of showing different group OLMs was helpful with participants commenting on the use of charts 1 and 3 for individual performance and charts 2, 4 and 5 for group performance. Comments: *P12– "It gives good ideas of how their process was, and this is good for feedback which is important and it also gives a summary of what we did, and the graphs are cool to look at"*; *P13– "Users might be interested to see how they performance and if they worked together, self-reflection is really useful"*; and *P14– "It can tell users a lot of information and may help them next time and [identifying] who is least active might be encouraging to try to do better"*.

7 Conclusion

We built a series of OLM visualisations for the purpose of analysing whether individuals could understand group and individual processes in order to support reflection in group brainstorming. Results showed learners found the OLMs relatively easy to comprehend and were able to answer our core questions. In the process of the study, we learnt which visualisations were most commonly referred to and why, leading to a greater understanding of the importance of different views for reflection. Our future work will be to build this into our tabletop brainstorming system, and show the visualisations through a scripted approach, to determine the effects of the OLMs when in real use.

References

1. Al-Qaraghuli, A., Zaman, H., Olivier, P., Kharrufa, A., Ahmad, A.: Analysing tabletop based computer supported collaborative learning data through visualization. Visual Informatics: Sustaining Research and Innovations pp. 329–340 (2011)

2. Bull, S., Britland, M.: Group Interaction Prompted by a Simple Assessed Open Learner Model that can be Optionally Released to Peers. In: Proceedings of Workshop on Personalisation in E-Learning Environments at Individual and Group Level, User Modeling (2007)
3. Bull, S., Vatrapu, R.: Supporting collaborative interaction with open learner models: Existing approaches and open questions. Computer Supported Collaborative Learning (2011)
4. Bull, S., Kay, J.: Student Models that Invite the Learner In: The {SMILI} Open Learner Modelling Framework. IJAIED, International Journal of Artificial Intelligence 17(2), 89–120 (2007)
5. Bull, S., Kay, J.: Metacognition and Open Learner Models. In: The 3rd Workshop on Meta-Cognition and Self-Regulated Learning in Educational Technologies, at ITS2008 (2008)
6. Clayphan, A., Collins, A., Ackad, C., Kummerfeld, B., Kay, J.: Firestorm: a brainstorming application for collaborative group work at tabletops. In: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces. pp. 162–171. ACM (2011)
7. Clayphan, A., Kay, J., Weinberger, A.: Enhancing brainstorming through scripting at a tabletop. In: Educational Interfaces, Software, and Technology 2012: 3rd Workshop on UI Technologies and Educational Pedagogy (2012)
8. Clayphan, A., Kay, J., Weinberger, A.: ScriptStorm: scripting to enhance tabletop brainstorming. In: Special Issue on EIST in Personal and Ubiquitous Computing (PUC) Journal (to appear) (2013)
9. Diehl, M., Stroebe, W.: Productivity loss in brainstorming groups: Toward the solution of a riddle. Journal of Personality and Social Psychology; Journal of Personality and Social Psychology 53(3), 497 (1987)
10. Do-Lenh, S.: Supporting Reflection and Classroom Orchestration with Tangible Tabletops. Ph.D. thesis, École Polytechnique Fédérale De Lausanne (2012)
11. Isaksen, S.: A review of brainstorming research: Six critical issues for inquiry. Creative Research Unit, Creative Problem Solving Group-Buffalo (1998)
12. Mabbott, A., Bull, S.: Alternative views on knowledge: presentation of open learner models. In: Intelligent Tutoring Systems. pp. 131–150. Springer (2004)
13. Martínez, R., Collins, A., Kay, J., Yacef, K.: Who did what? who said that?: Collaid: an environment for capturing traces of collaborative learning at the tabletop. In: Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces. pp. 172–181. ACM (2011)
14. Martinez, R., Wallace, J., Kay, J., Yacef, K.: Modelling and identifying collaborative situations in a collocated multi-display groupware setting. In: Artificial Intelligence in Education. pp. 196–204. Springer (2011)
15. Martinez Maldonado, R., Kay, J., Yacef, K., Schwendimann, B.: An Interactive Teacher's Dashboard for Monitoring Groups in a Multi-tabletop Learning Environment. In: Intelligent Tutoring Systems. pp. 482–492. Springer (2012)
16. Osborn, A.: Applied Imagination, principles and procedures of creative thinking. Scribner's (1953)
17. Schon, D.: The reflective practitioner: How professionals think in action. Basic Books (1983)
18. Wang, H., Li, T., Rosé, C., Huang, C., Chang, C.: Vibrant: A brainstorming agent for computer supported creative problem solving. In: Intelligent Tutoring Systems. pp. 787–789. Springer (2006)

Encouraging Online Student Reading with Social Visualization Support

Julio Guerra, Denis Parra, Peter Brusilovsky

School of Information Sciences
University of Pittsburgh
130 N Bellefield Ave, Pittsburgh, PA, 15260, USA
{jdg60,dap89,peterb}@pitt.edu

Abstract. In this paper, we describe ReadingCircle, a system designed to explore an alternative approach to encouraging reading among students. It is based on recent research on open student modeling, social comparison and social visualization. The idea of this approach is to develop social visualization of students' reading progress. The visualization will reveal such reading progress through several levels (from chapters to sections to pages) and allow students to visually compare their progress with both the class as a whole and individual peers.

Keywords. elearning, online reading, social visualization, social comparison, open student model

1 Introduction

Almost every college course requires students to complete weekly readings from course textbooks or other course materials, an effort critical to the students' success in the course. However, it is not easy for an instructor to determine whether or not the students have in fact completed the assigned readings. To combat this trend, instructors have to implement various approaches to encourage student reading and to ensure that reading assignments are completed. In smaller classes, these approaches could be both creative and efficient – such as group discussions. In larger classes, however, instructors find it difficult to assess the students' progress on the readings in an efficient way. Contemporary approaches such as randomly surveying students in class or administering pop-quizzes are neither creative nor efficient. Also, reading assignments produce no artifacts to grade by. As a result, the students frequently are not motivated to complete the reading assignments.

In this paper, we describe *ReadingCircle*, an alternative approach to encouraging student reading that is based on our recent research combining open student modeling, social comparison and social visualization [1]. The premise of this approach is to engage social visualization of student reading progress as a barometer of progress. The visualization exhibits progress on several levels (from chapters to sections to pages), and allows the students to visually compare their progress with both the class as a whole and individual peers. We expect that social progress visualization will improve student awareness of readings left to do and class progress; the ultimate goal is to encourage students to do more readings. This paper presents our motivation for designing and creating this social reading application.

2 Related Work

Social Comparison. According to social comparison theory [7], people tend to compare their achievements and performance with others who are similar to them in some way. Earlier social comparison studies [11] demonstrated that students were inclined to select the more challenging tasks because of being exposed to social comparison conditions. Later studies showed that social comparison decreases social loafing and increases productivity by reinforcing good behavior through a graphical feedback tool [9]. A synthesis review of social comparison studies' summarized that applying social comparison in the classroom often leads to better student performance [8].

Social Visualization in E-Learning. The visual approach is a common technique to represent or organize data about multiple students in an informative way. For instance, social navigation, which is a set of methods for organizing users' explicit and implicit feedback to support information navigation [5], leverages the social phenomenon where people tend to follow the "footprints" of other people [2]. The educational value of social navigation have been confirmed in several studies [3, 6]

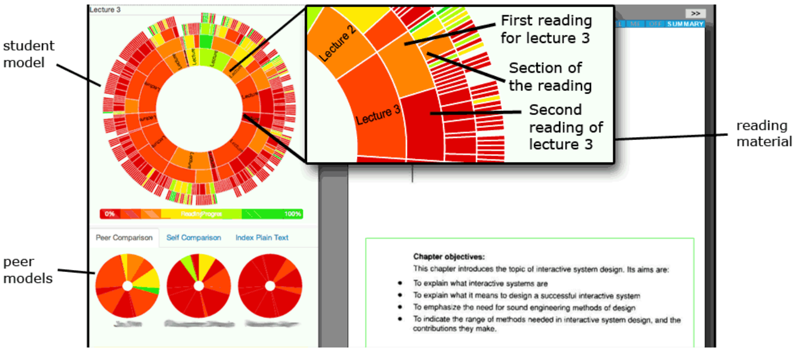
It is common to provide learners with the average values of the group model through social visualization in E-Learning; such as the average knowledge of the group on a given topic. Vassileva and Sun [10] investigated community visualization in online communities. They opined that social visualization increases social interaction among students, encourages competition, and offers students the opportunity to build trust in others and in the group. Bull & Britland [4] showed that releasing the models to their peers increases the discussion among students and encourages them to start working with learning content sooner.

In our prior work [1] we combined social visualization with open student modeling visualization to provide students with a holistic and easy-to-grasp view of their progress on answering java programming questions, and at the same time, allowing them to compare their progress with that of other students in the class. Our classroom studies demonstrated that the social visualization interface provided a remarkable increase in student work with problems. It also demonstrated that a circular design provides a better approach than a tree map to show progress over hierarchically structured content. This paper extends this work and presents a *social progress visualization interface to support online reading*. This interface takes advantage of some of the successful design ideas from our previous projects, and aims to work with a very different type of content. We expect that the new interface will provide clear guidance to the students to manage their reading process and to significantly increase their motivation to read.

3 The ReadingCircle Interface

The main challenge in our social reading interface design was to combine a simple social progress visualization of student progress over a flat list of topics (our past interface works with topics in Java) with a more complicated and hierarchical structure of student reading assignments. In addition, we wanted to employ the visualization not only as a social comparison tool, but also as a social navigation tool that provides orientation support and navigation support for a large body of assigned readings.

In light of these goals, the ReadingCircle system interface is divided into a social navigation component and a reading element as can be seen in Figure 1. The reading part on the right shows the current reading material and allows the student to make annotations and see annotations from peers. The social navigation component on the left aims to present visually the open student and peer models. The visualization of the student model (the top right part in Figure 1) is also the main content navigation control. We chose this circular shape approach because it requires less space to show the whole (hierarchical) content structure.



Chapter objectives:

This chapter introduces the topic of interactive system design. Its aims are:

- To explain what interactive systems are
- To explain what it means to design a successful interactive system
- To emphasize the need for sound engineering methods of design
- To indicate the range of methods needed in interactive system design, and the contributions they make.

Figure 1. The ReadingCircle interface. The left part shows the student model (top) and peer models (bottom). The material is shown on the right side. A small portion of the user model is magnified at top center.

The circular shaped model presents the content structure of a course, organized clockwise, of 13 lectures. Each lecture consisted of one or more readings which can be chapters or sections from several books used in the course. Following the hierarchical structure of the reading (for example, a chapter has sections, and sections has subsections), the sector in the visualization corresponding to the reading is "opened" to reveal the fine-grained content. The top center rectangle in Figure 1 presents a closer view of the third lecture (lecture 3). By clicking in each sector, the student is presented with a menu of the related content displayed in the right side. The color of the sections indicates the progress on a scale ranging from red (not seen) to green (completed). The progress is computed by aggregating the evidence of the user reading each terminal subsection to upper level subsections, chapter and lectures. We track the individual page loads (i.e. the individual pages of each reading), and the actions (clicks, annotations) of the user in the reader interface. The bottom part of the left side in Figure 1 presents 3 tabs: Peer Comparison, Self Comparison and Index Plain Text. The Peer Comparison tab shows thumbnail models of three peers. The models display only the lecture level. The Self-Comparison tab is similar and shows three previous models of the current student (over the past 3 weeks). We aim to explore the effect of self-comparison as we study peer comparison.

The social reading interface presented above is currently going through a classroom study in a large graduate class. Using log analysis and questionnaires, we hope to assess the impact of, and the student attitude towards, the tool.

4 References

1. Hsiao, I-H., Guerra, J. , Parra, D., Bakalov, F., König-Ries, B, and Brusilovsky, P. Comparative social visualization for personalized e-learning. In Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12). ACM, New York, NY, USA, 303-307 (2012).
2. Brusilovsky, P., Chavan, G., and Farzan, R. Social adaptive navigation support for open corpus electronic textbooks. In: Proc. of Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2004), Springer-Verlag, 24-33 (2004)
3. Brusilovsky, P., Sosnovsky, S., and Yudelson, M. Addictive links: The motiva-tional value of adaptive link annotation. New Review of Hypermedia and Multimedia, 15, 1 (2009), 97-118.
4. Bull, S. and Britland, M. Group Interaction Prompted by a Simple Assessed Open Learner Model that can be Optionally Released to Peers. In Proc. of Workshop on Personalization in E-learning Environments at Individual and Group Level at the 11th International Conference on User Modeling, UM 2007, 24-32 (2007).
5. Dieberger, A., Dourish, P., Höök, K., Resnick, P., and Wexelblat, A. Social navigation: Techniques for building more usable systems. interactions, 7, 6 (2000), 36-45.
6. Farzan, R. and Brusilovsky, P. AnnotatEd: A social navigation and annotation service for web-based educational resources. New Review in Hypermedia and Multimedia, 14, 1 (2008), 3-32.
7. Festinger, L. A theory of social comparison processes. Human Relations, 7 (1954), 117-140.
8. Pieternel Dijkstra, Hans Kuyper, Greetje van der Werf, and, A.P.B., and Zee, Y.G.v.d. Social Comparison in the Classroom: A Review. REVIEW OF EDUCATIONAL RESEARCH, 78, 4 (2008).
9. Shepherd, M.M., Briggs, R.O., Reinig, B.A., Yen, J., and Jay F. Nunamaker, J. Invoking social comparison to improve electronic brainstorming: beyond anonymity. J. Manage. Inf. Syst., 12, 3 (1995), 155-170.
10. Vassileva, J. and Sun, L. Evolving a Social Visualization Design Aimed at Increasing Participation in a Class-Based Online Community. International Journal of Cooperative Information Systems (IJCIS), 17, 4 (2008), 443-466.
11. Veroff, J. Social comparison and the development of achievement motivation., New York: Sage (1969).

Academically Productive Talk: One Size Does Not Fit All

David Adamson & Carolyn P. Rosé

Language Technologies Institute, Carnegie Mellon University, Pittsburgh PA, 15213
{dadamson,cprose}@cs.cmu.edu

Abstract. We present a study in which we experimentally manipulate the form of support offered to groups of three students during collaborative learning. Specifically, we contrast two forms of Academically Productive Talk (APT) facilitation, known as Revoicing and Agree-Disagree. The first form has been demonstrated effective with the target age group (i.e., 9th grade) on an earlier more difficult unit. The second form has been demonstrated effective with older kids. Results suggest that with this age group, facilitation with Revoicing may be more effective than Agree-Disagree. Implications for future work are discussed.

Keywords: dynamic support for collaborative learning, academically productive talk, discussion for learning.

1 Introduction

Collaborative learning activities, when delivered effectively, can provide significant cognitive, metacognitive, and social benefits to students [18][32][35]. Studies in the field of computer-supported collaborative learning have demonstrated the pedagogical value of social interaction [37][38]. Prior work on adaptive support for collaborative learning has adapted hint-based support originally developed for individual learning to support peer tutoring [13], and other work has grown out of earlier efforts to develop tutorial dialogue agents originally designed for individual learning [16][30][40][41]. This form of dynamic agent-based support for collaborative learning was historically tailored to specific learning populations and content domains [22], which limits its generality. More generalizable forms of support would increase the potential for impact, but as we discuss in this paper, raise new questions about principles for adaptation that would enable us as system developers to provide solutions that can be effective for diverse student populations.

Our recent efforts are in the direction of intelligent conversational agents acting as discussion facilitators, offering support behaviors that are not tied to a particular content-area or context [1][10][14]. The design of such support is in line with the literature on facilitation of collaborative learning groups [17]. In particular, it draws upon a body of work that has shown that certain forms of classroom discussion facilitation, termed Accountable Talk, or Academically Productive Talk (APT), are beneficial for learning with understanding [3][8][9][28][29][33][34][39].

In this paper we present results from a study in which we contrast two forms of APT based support. The first form, Revoicing support, has been found in prior work to achieve positive learning effects with the target student population of 9th graders [14] on an earlier and more difficult lesson. The other form of support, Agree-Disagree support, has been found to be effective with older, more advanced learners [1] in a different content domain. In this study, we show that with a 9th grade student population, Revoicing support is slightly more effective that Agree-Disagree support. These results contribute towards an empirical foundation for adapting APT based support to differences in content domain difficulty and differences in the developmental stage of target learners.

In the remainder of the paper we first review the state of the art in agent based support for collaborative learning. Next we describe two forms of APT-based support. Then we describe an evaluation study where we compare the effectiveness of these two forms of support for 9th grade biology students working on a genetics unit that is relatively easy for them. We conclude with discussion of results and future directions.

2 Prior Work

Academically Productive Talk has grown out of frameworks that emphasize the importance of social interaction in the development of mental processes. Michaels, O'Connor and Resnick [26] describe a number facilitating moves that teachers can employ to promote student-centered classroom discussion. A selection of these moves are presented in Table 1. In studies where teachers used similar facilitation strategies, students showed dramatic improvement on standardized math scores, transfer to reading test scores, and retention of transfer for up to 3 years [8][9].

Table 1. Selected Accountable Talk Moves

APT Move	**Example**
Revoicing a student's statement	"So, let me see if I've got your thinking right. You're saying XXX?"
Asking students to apply their own reasoning to someone else's reasoning	"Do you *agree or disagree*, and why?"

Collaboration scripts are a common way to describe and structure support for collaborative learning [20] within the field of computer-supported collaborative learning. A collaboration script may describe any of a wide range of features of collaboration scenarios, including the tasks, timing, roles, and the methods and desired patterns of interaction between the participants. A script can describe the collaborative activity at the macro or micro level [12]. Macro-scripts describe the sequence and structure each phase of a group's activities, specifying coarse-grained features such as assigned tasks and roles, and the overall shape of the activity. Micro-scripts, on the other hand, are models of dialogue and argumentation embedded in the activity, and are intended to be adopted and progressively internalized by the participants [19]. Micro-scripts can be realized by sharing prompts or hints with the user, guiding or providing models for their contributions [36]. While traditional collaboration scripts such as these can provide some degree of support for conversational and reasoning practices, they fall short of delivering the active, engaged facilitation described by the APT literature.

In particular, such scripts are static, and do not respond to changes in (or awareness of) student need or ability during the activity. Such non-adaptive approaches risk detrimental over-scripting [11]. More preferable would be the delivery or adjustment of supports in response to the automatic analysis of student activity [2][31]. The collaborative conversational agents described by Kumar and Rosé [24] were among the first to implement such dynamic scripting in a CSCL setting, with demonstrable gains over otherwise equivalent static support. Likewise, recent work by Baghaei et al [6] and Diziol et al [13] show that adaptive supports can have meaningful effects on student learning and interaction.

3 Dynamic Support for Academically Productive Talk

Two dynamic conversational supports based upon APT facilitation, namely Revoicing and Agree-Disagree, were implemented and evaluated in this study. The open-source Bazaar architecture [2] was used to author and orchestrate the conversational agent and the support behaviors described below.

3.1 Revoicing Support

One of the forms of support evaluated in this paper is a Bazaar component that performs an Academically Productive Talk move referred to as Revoicing. The agent compares student statements against a list of conceptually correct statements developed with teachers. In the study described in this article, 35 such statements were developed and validated against pilot data. For each student turn, we calculate a measure of “bag of synonyms” cosine similarity against each expert statement, based on the method described by Fernando and Stevenson [15]. If this similarity value exceeds a conservatively high threshold, we consider the student's turn to be a possible paraphrase of the matched statement, and thus “revoicable” (this threshold was determined through tests against pilot data, such that at least 80% of the revoicings suggested for candidate student were on-target). The Revoicing component may respond by offering the matched statement as a paraphrase of the student's turn, for example “So what I hear you saying is XXX. Is that right?” No statement may trigger a revoice move more than once.

3.2 Agree-Disagree Support

The other support we evaluate is a Bazaar component which performs the APT Agree-Disagree move. Candidate student statements are identified using the same method as described for the Revoicing support, but with a lower threshold that allows looser matches. After detecting such a candidate, the agent waits for the other students in the group to respond to it. If another student responds with an evaluation of their peer’s contribution (for example, “I agree” or “I think you’re wrong”, as recognized by a small list of hand-crafted regular expressions), but doesn’t support the evaluation with an explanation, the agent will encourage this second student to provide one. If a student instead follows up with another APT candidate statement, the agent does nothing, leaving the floor open for productive student discussion to continue unimpeded, reducing the risk of over-scripting their collaboration. If the other students do not respond with either an evaluation or a contentful follow-up, the agent prompts them to comment on the candidate statement – for example, “What do you think about Billy’s idea? Do you agree or disagree?”

4 Method

Following the literature on APT used as a classroom facilitation technique, in this study we test the hypothesis that appropriate APT support in a computer-supported collaborative learning setting will both intensify the exchange of reasoning between students during the collaborative activity, and increase learning during the activity.

4.1 Instructional Content and Study Procedure

Participants: This study was conducted in seven 9th grade biology classes of an urban school district. The classes were distributed across two teachers (with respectively 3 and 4 classes) for a total of 143 students total, with 76 consenting. Students were randomly assigned to groups of 3. Groups were randomly assigned to conditions. Only data from consenting students was used in the analysis presented here.

Experimental Manipulation: This study was run as a 3 condition between subjects design in which the APT agents provided some behaviors in common across conditions, but other behaviors were manipulated experimentally. Across all conditions, the agent provided the same macro level support by guiding the students through the activity using the same phases introduced in such a way as to control for time on task. It was the micro-scripting behaviors that were manipulated experimentally in order to create the three conditions of the design. The first experimental conditions was Revoicing, using the behavior described above. The second was the Agree-Disagree condition, where the Agree-Disagree behavior discussed above was used. In the control condition, neither of these behaviors was used.

Learning Content: The study was carried out during a module introducing the concepts of genetics, heredity, and single-trait inheritance. In the activity, student groups were presented with a set of three problems and asked to reason about the physical and genetic traits of the likely parents of a set of siblings. Specifically, in each problem, students were shown a litter of eight kittens that varied in fur color (either orange or white), and were instructed to identify the genotypes and phenotypes of the parents, and to explain their reasoning to their teammates. This sort of “backwards” reasoning had not been explicitly addressed in the course to date – students only had prior experience with “forward” reasoning from given parental traits. The mystery parents were presented as the inputs to an unpopulated Punnett square, as shown in Figure 2. As an incentive, students were told that the best team, determined by a combination of discussion quality and post-test scores, would be awarded a modest prize. Each of the three tasks was progressively harder than the last in that fewer clues about the parent's identities were included. The collaborative task content, the macro-scripts that supported it, and the list of statements powering the APT support were all developed iteratively with feedback from teachers and content experts.

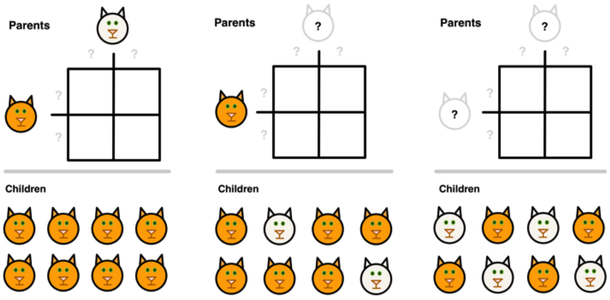
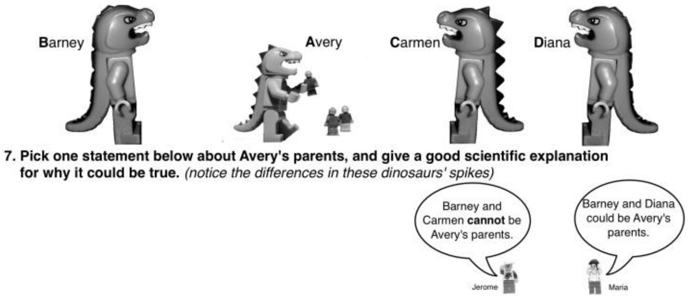


Fig. 1. Task sequence for the collaborative activity.

Study Procedure: The study was conducted over three phases, which occurred as single class periods over two school days. The first phase ("day 1") involved the teachers taking a pre-test at the end of a regular class session.

The second phase ("day 2") was centered around a 20 minute collaborative computer-mediated activity during which the experimental manipulation took place. The students performed the activity in groups of three, scaffolded by a conversational agent. Students within classes were randomly assigned to groups, then groups to conditions. The activity was introduced by a cartoon handout depicting the use of APT, and a ten-minute presentation describing the task and reviewing the basics of genetics and heredity. At the end of this second phase, the students took a post-activity test.

The computer activity was intended to equip the students with enough empirical data and attempts at reasoning to prepare them for the third phase ("day 3"), a full class APT discussion with their teacher, during which they would reconcile their different understandings and explanations. At the end of this discussion, they took a post-discussion test.


Barney
Avery
Carmen
Diana
Barney and Carmen **cannot** be Avery's parents.
Barney and Diana could be Avery's parents.
Jerome
Maria

7. Pick one statement below about Avery's parents, and give a good scientific explanation for why it could be true. *(notice the differences in these dinosaurs' spikes)*

Fig. 2. Concept cartoon question from the post-activity test.

4.2 Measurement

Domain knowledge was measured at three time points using a paper based test. Each of the three tests (Pre-Test, Post-Activity Test, Post-Discussion Test) followed a similar format: a set of multiple choice problem-solving questions addressing forward and backward reasoning about single inheritance, and what we refer to as a *concept cartoon*, in which a set of potential parents for a single child was displayed, along with two hypotheses for who the child's parents might be. Students were instructed to select one hypothesis and clearly explain the conditions that would allow it to be true – either hypothesis could be correct, with different underlying assumptions. Student responses were graded with a rubric assessing the quality and depth of their explanation, including explicit displays of reasoning.

Each test covered the same knowledge but used different scenarios. The knowledge to be covered by each test was established in coordination with the teachers, with teacher trainers who identified common misconceptions, and with test results from a study run with the same content the previous year. After an initial round of consensus grading by two graders on a subset of the tests to establish a scoring guide, the remaining tests were divided and scored by one grader each.

Table 2. Total test scores (standard dev) for Pretest, Post-Activity Test, and Post-Discussion Test in the 3 conditions.

	Control	Revoice	Agree-Disagree
Pretest	5.5 (3.1)	5.5 (3.2)	3.9 (3.0)
Post-Activity Test	6 (3.4)	6.3 (3.1)	4 (3.1)
Post-Discussion Test	5.7 (3.1)	6.1 (2.9)	4.8 (3.3)

4.3 Results

First we tested whether students learned during the online activity. Test scores were divided into explanation questions and problem solving questions. Thus, for each test, each student has two scores. In order to evaluate learning, we used an ANOVA with Test Score as the dependent variable, Explanation vs Skill, Pretest vs Post-Activity Test, Condition, and Teacher as independent variables. We added Teacher as a variable because we noticed that students from one teacher learned significantly more than students from the other teacher. In this analysis, all of the independent variables were significant except Pre-test vs Post-test, which was marginal, $F(1, 270) = 3.6$, $p < .06$. There were no significant interactions between independent variables. Thus we find qualified evidence that students learned during the online activity, across conditions. However, on inspecting the average scores in Table 1, we see barely any evidence of learning in the Agree-Disagree condition. The most learning we see is about .25 standard deviations in the Revoicing condition, and about half that in the Control condition.

We also tested whether students learned during the Post-activity discussion. In this case, when comparing between the Post-Activity test and the Post-Discussion test there was no significant difference. In fact, the trend was that students scored more poorly on the Post-Discussion test than the Post-Activity test, except in the Agree-Disagree condition, where the students came into the discussion with less knowledge than students in the other two conditions, and seemed to be able to use the Post-activity Discussion to catch up, which is consistent with findings from earlier studies (Dyke et al., in press).

We compared learning across conditions between Pre-test and Post-Activity test, and between Pre-test and Post-Discussion test. In both cases, we used an ANCOVA with the posttest measure (i.e., Post-Activity test in the first comparison and Post-Discussion test in the second) as the dependent variable and the Pre-test as the covariate. We retained the Teacher variable in addition to the condition variable. In neither case do we find a significant effect of condition. However between the Pre-test and Post-activity test the trend is for adjusted posttest scores to be higher than the control condition in the Revoicing condition (by .13 standard deviations) and lower than the control condition in the Agree-Disagree condition (by .4 standard deviations), with very similar trends when comparing between Pre-test and Post-Discussion test.

We acknowledge that stronger claims could be made by conducting our analysis using multilevel modeling. However, such complex modeling techniques require larger data sets in order to avoid falling prey to type II errors during hypothesis testing. Due to the small size of our data, we employed simpler methods for our analysis.

5 Discussion & Conclusions

Overall, the results are weak. However, the results suggest a differential effect of the two experimental conditions. The trend in favor of the Revoicing condition is consistent with earlier studies with the same age group, but on a more difficult unit in the course [14]. The trend to learn less than the control condition in the Agree-Disagree condition is in contrast to earlier results with more advanced learners [1] where students in the Agree-Disagree condition learned significantly more than in the control condition. These suggestive results will need to be followed up with additional experimentation before we can have more confidence in the findings. However, they do suggest that the effect of these APT facilitation strategies on learning depend on the difficulty of the unit and the developmental stage of the learners, and that more results are needed to inform effective strategies for supporting groups of learners.

6 Acknowledgements

This work was supported in part by NSF grant SBE 0836012 to the Pittsburgh Science of Learning Center.

7 References

[1] Adamson, D., Ashe, C., Jang, H., Yaron, D., & Rosé, C. (2013). Intensification of Group Knowledge Exchange with Academically Productive Talk Agents. *Proceedings of the 10th International Conference on Computer Supported Collaborative Learning*, Madison Wisconsin, July 2013.

[2] Adamson, D., & Rosé, C. (2012). Coordinating multi-dimensional support in collaborative conversational agents. In *Proceedings of Intelligent Tutoring Systems* (pp. 346-351). Springer Berlin/Heidelberg.

[3] Adey, P., & Shayer, M. (1993). An Exploration of Long-Term Far-Transfer Effects Following an Extended Intervention Program in the High School Science Curriculum. *Cognition and Instruction, 11*(1), pp 1-29.

[4] Ai, H., Kumar, R., Nguyen, D., Nagasunder, A., Rosé, C. P. (2010). Exploring the Effectiveness of Social Capabilities and Goal Alignment in Computer Supported Collaborative Learning, in *Proceedings of Intelligent Tutoring Systems*.

[5] Azmitia, M. & Montgomery, R. (1993). Friendship, transactive dialogues, and the development of scientific reasoning, *Social Development* 2(3), pp 202-221.

[6] Baghaei, N., Mitrovic, A., & Irwin, W. (2007). Supporting collaborative learning and problem solving in a constraint based CSCL environment for UML class diagrams, *International Journal of Computer Supported collaborative Learning* 2(3), pp 159-190.

[7] Berkowitz, M., & Gibbs, J. (1983). Measuring the developmental features of moral discussion. *Merrill-Palmer Quarterly*, 29, pp 399-410.

[8] Bill, V. L., Leer, M. N., Reams, L. E., & Resnick, L. B. (1992). From cupcakes to equations: The structure of discourse in a primary mathematics classroom. *Verbum*, 1, 2, 63-85.

[9] Chapin, S., & O'Connor, C. (2004). Project challenge: Identifying and developing talent in mathematics within low-income urban schools. *Boston University School of Education Research Report* (Vol. 1, pp. 1-6).

[10] Clarke, S., Chen, G., Stainton, K., Katz, S., Greeno, J., Resnick, L., Howley, H., Adamson, D., Rosé, C. P. (2013). *The Impact of CSCL Beyond the Online Environment, Proceedings of Computer Supported Collaborative Learning*

[11] Dillenbourg, P. (2002). Over-scripting CSCL : The risks of blending collaborative learning with instructional design . *Three worlds of CSCL: Can we support CSCL*, pp. 61-91.

[12] Dillenbourg, P., Hong, F. (2008). The mechanics of CSCL macro scripts. *International Journal of Computer-Supported Collaborative Learning* 3(1), pp 5-23.

[13] Diziol, D., Walker, E., Rummel, N., & Koedinger, K. R. (2010). Using intelligent tutor technology to implement adaptive support for student collaboration. Educational Psychology Review, 22(1), 89-102.

[14] Dyke, G., Adamson, D., Howley, I., Rosé, C.P. (Under Review). Enhancing Scientific Reasoning and Explanation Skills with Conversational Agents, submitted to *IEEE Transactions on Learning Technologies*.

[15] Fernando, S., and Stevenson, M. (2008). A semantic similarity approach to paraphrase detection, *Computational Linguistics UK (CLUK 2008) 11th Annual Research Colloquium*.

[16] Graesser, A., VanLehn, K., the TRG, & the NLT (2002). Why2 Report: Evaluation of Why/Atlas, Why/AutoTutor, and Accomplished Human Tutors on Learning Gains for Qualitative Physics Problems and Explanations. *LRDC Tech Report, University of Pittsburgh*.

[17] Hmelo-Silver, C. E. & Barrows, H. S. (2006). Goals and Strategies of a Problem-based Learning Facilitator. *The Interdisciplinary Journal of Problem Based Learning*, 1(1), pp 21-39.

[18] Kirschner, F., Paas, F., & Kirschner, P. A. (2009). A cognitive load approach to collaborative learning: United brains for complex tasks. *Educational Psychology Review*, 21, 31–42.

[19] Kobbe, L., Weinberger, A., Dillenbourg, P., Harrer, A., Hämäläinen, R., Häkkinen, P., Fischer, F. (2007). Specifying computer-supported collaboration scripts. *The International Journal of Computer-Supported Collaborative Learning* 2(2-3), pp 211-224.

[20] Kollar, I., Fischer, F., Hesse, F.W. (2006). Collaborative scripts - a conceptual analysis. *Educational Psychology Review* 18(2), pp 159-185.

[21] Kumar, R., Gweon, G., Joshi, M., Cui, Y., Rosé, C. P. (2007a). Supporting Students Working Together on Math with Social Dialogue. *Proceedings of the SLaTE Workshop on Speech and Language Technology in Education*

[22] Kumar, R., Rosé, C. P., Wang, Y. C., Joshi, M., Robinson, A. (2007b). Tutorial Dialogue as Adaptive Collaborative Learning Support, *Proceedings of Artificial Intelligence in Education*.

[23] Kumar, R., Ai, H., Beuth, J., Rosé, C. P. (2010). Socially-capable Conversational Tutors can be Effective in Collaborative Learning Situations, in *Proceedings of Intelligent Tutoring Systems*.

[24] Kumar, R., Rosé, C.P. (2011). Architecture for Building Conversational Agents that Support Collaborative Learning. *IEEE Transactions on Learning Technologies* 4(1).

[25] Lison, P. (2011). Multi-Policy Dialogue Management. In *Proceedings of the SIGDIAL 2011 Conference*, Association for Computational Linguistics, pp. 294-300

[26] Michaels, S., O'Connor, C., & Resnick, L.B. (2007). Deliberative discourse idealized and realized: Accountable talk in the classroom and in civic life. *Studies in Philosophy and Education*.

[27] Rabe-Hesketh, S., Skrondal, A., & Pickles, A. (2004). *GLLAMM Manual*. University of California, Berkely. U. C. Berkeley Division of Biostatistics Working Paper Series, Paper 160.

[28] Resnick, L. B., Asterhan, C. A., & Clarke, S. N. (in press). Socializing Intelligence through Academic Talk and Dialogue. *Washington, DC: American Educational Reserach Association*.

[29] Resnick, L. B., Salmon, M., Zeitz, C. M., Wathen, S. H., & Holowchak, M. (1993). Reasoning in conversation. *Cognition and Instruction*, 11(3-4), 347-364.

[30] Rosé, C. P., Jordan, P., Ringenberg, M., Siler, S., VanLehn, K., Weinstein, A. (2001). Interactive Conceptual Tutoring in Atlas-Andes, *Proceedings of AI in Education*.

[31] Rosé, C. P., Wang, Y.C., Cui, Y., Arguello, J., Stegmann, K., Weinberger, A., Fischer, F. (2008). Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning. *The International Journal of Computer-Supported Collaborative Learning* 3(3), 237-271.

[32] Scardamalia, M., & Bereiter, C. (1993). Technologies for knowledge-building discourse. *Communications of the ACM*, 36(5), 37–41.

[33] Topping, K. J., & Trickey, S. (2007a). Collaborative philosophical enquiry for school children: Cognitive effects at 10-12 years. *British Journal of Educational Psychology*, 77(2), 271-288.

[34] Topping, K. J., & Trickey, S. (2007b). Collaborative philosophical inquiry for schoolchildren: Cognitive gains at 2-year follow-up. *British Journal of Educational Psychology*, 77(4), 787-796.

[35] Webb, N. M., & Palinscar, A. S. (1996). Group processes in the classroom. In D. C. Berliner & R. C. Calfee (Eds.), *Handbook of educational psychology* (pp. 841–873). New York: Prentice Hall.

[36] Wecker, C., Fischer, F. (2007). Fading scripts in computer-supported collaborative learning: The role of distributed monitoring. *Proceedings of the 8th iternational conference on Computer Supported Collaborative Learning*, pp. 764-772.

[37] Weinberger, A., Stegmann, K., & Fischer, F. (2007). Scripting argumentative knowledge construction: Effects on individual and collaborative learning. In C. Chinn, G. Erkens, & S. Puntambekar (Eds.), *Mice, minds, and society: CSCL 2007* (pp. 37-39). New Brunswick, NJ: International Society of the Learning Sciences.

[38] Weinberger, A., Fischer, F. (2006). A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Computers & Education* 46(1), 71–95.

[39] Wegerif, R., Mercer, N., & Dawes, L. (1999). From social interaction to individual reasoning: an empirical investigation of a possible socio-cultural model of cognitive development. *Learning and Instruction*, 9(6), 493-516.

[40] Wiemer-Hastings, P., Graesser, A., Harter, D. and the Tutoring Research Group, (1998). The Foundations and Architecture of AutoTutor. In B. Goettl, H. Halff, C. Redfield & V. Shute (Eds.) *Intelligent Tutoring Systems: 4th International Conference* (ITS '98) (pp334-343). Berlin: Springer-Verlag.

[41] Zinn, C., Moore, J. D., & Core, M. G. (2002). A 3-Tier Planning Architecture for Managing Tutorial Dialogue. In S. A. Cerri, G. Gouardères & F. Paraguaçu (Eds.) *Proceedings of the Sixth International Conference on Intelligent Tutoring Systems, ITS 2002* (pp. 574-584). Berlin: Springer Verlag.

Towards Supporting 'Learning To Learn Together' in the Metafora platform

Manolis Mavrikis[1], Toby Dragon[2], Nikoleta Yiannoutsou[3], Bruce M. McLaren[2,4]

[1]London Knowledge Lab, Institute of Education, University of London, WC1H 0AL, UK,
m.mavrikis@lkl.ac.uk
[2]CeLTech, Saarland University, Saarbruecken 66123, Germany,
toby.dragon@celtech.de
[3]Educational Technology Lab, National and Kapodistrian University of Athens, Greece 106 79,
nyiannoutsou@ppp.uoa.gr
[4]Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA, USA
bmclaren@cs.cmu.edu

Abstract. Computer-Supported Collaborative Learning (CSCL) has been demonstrated to improve student interaction in complex collaborative learning scenarios. When orchestrated appropriately, it also provides opportunities for learning high-level social learning skills, or "learning to learn together" (L2L2), but these opportunities are often only dealt with implicitly. This paper presents work towards an intelligent system that can scaffold L2L2 across many domains by (a) offering carefully-designed message templates that encourage peers to communicate with their groups about their learning process, (b) analyzing student work and recommending a specific set of these message templates that are pertinent to their moment-by-moment interaction. We present methods by which the system can use automated analysis techniques to recognize opportunities where students might benefit from these messages, and either send the message directly or prioritize message templates for students' use.

Keywords. Computer-Supported Collaborative Learning, Exploratory Learning Environments, Learning to Learn Together, Intelligent Support of Social Interaction

1 Introduction

The Computer-Supported Collaborative Learning (CSCL) field has demonstrated that success in complex collaborative environments depends on several factors including the type of task, the learning scenario, and the collaborative skills of the students involved. When orchestrated appropriately, these types of learning scenarios provide opportunities both for domain related learning and social meta-learning, or 'learning to learn together' (L2L2). However, for this to be possible, students and teachers alike need tools to elevate their conversation beyond solely subject matter, to recognize and practice high-level collaborative learning skills (L2L2 skills) in tandem with domain skills. Beyond simply providing appropriate interaction spaces, one of the goals behind CSCL and AI in Education systems is to provide more structured guidance. This type of automated support, however, is challenging in these complex scenarios, due to the variance in domains, learning scenarios, and intricacies of interrupting collaborative learning processes at appropriate points in time. All of these considerations limit the applicability of standard techniques for providing direct feedback.

With these challenges in mind, we argue for a broader perspective on the role of both feedback and AI in such scenarios. As discussed in [1], feedback in collaborative settings can manifest in many different ways, rather than limiting intervention mechanisms to messages flowing directly from an AI analysis system to individuals. We suggest a design where students, teachers (or facilitators in general), and automated agents can all offer feedback to individuals and groups, with the support of the system. Thus, the system takes on an additional role of providing tools and scaffolding to help students offer feedback to each other (a more indirect presentation of feedback). This scaffolding can be provided through *message templates*, generic phrases that focus attention on L2L2 concepts and can be tailored to fit the specific scenario at the time of use. These message templates are available in an intuitive and easy-to-use tool that enables students to send messages to one another, or for teachers to send messages to students. Utilizing this functionality, the AI system can go beyond the traditional role (i.e., direct presentation of feedback messages), to also scaffold the users in sending messages to each other (indirect presentation). To accomplish this, the AI system can recommend the most relevant message templates at any given point in time. Key questions to address when taking this approach include:

- What kinds of messages are most likely to promote L2L2 within task-focused group work?
- How can an intelligent system be developed to understand and identify when these messages might be most effective?
- How should the system deliver these messages or encourage the users to deliver them?

The rest of the paper is organized as follows. We first describe the context where this work is situated, in particular the Metafora platform and pedagogy that is being developed in the EU-funded Metafora project [2]. In Section 2, we briefly present the system and the key components of L2L2 that it is designed to help students develop and practice. In Section 3, we describe our process for developing appropriate, generally-applicable messages, and how these so-called ‘message templates’ are made generic and available for use within the system through techniques that allow the system to recognize and automatically respond to L2L2 behaviors. To conclude, we discuss our initial findings and future plans with respect to evaluating the approach.

2 Background and relevant work

2.1 The Metafora system and project

To support the L2L2 process, the Metafora project designed a platform that includes a planning tool designed for explicating and reflecting on the group learning process. Additionally, the platform contains the LASAD discussion environment [3] for developing arguments or discussions around the topics that emerge during the collaborative process. Of course, teaching these higher-level learning skills cannot be done without grounding the work with genuinely challenging tasks that require critical thinking skills (c.f. [4],[5]). The Metafora system offers a broad range of such learning activities across math and science by providing a suite of exploratory learning environments (microworlds and simulations). All of these tools are brought together in the Metafora platform, which serves both as a toolbox and as a communication architecture to support cross-tool interoperability. As a toolbox the system provides a graphical container in which the diverse learning tools can be launched and used (the Figures in Table A.1 give an impression of the Metafora system with the platform parts on the top and left borders and the graphically integrated tools in the main panel from center to right).

2.2 L2L2 in Metafora

The Metafora platform and tools have been designed and implemented to provide support for key components of L2L2, defined through both literature review and design-based research. In the interest of space we refer the reader to ([2]; [6]) and the project deliverables (see http://www.metafora-project.org) but in brief the four L2L2 aspects are as follows:

- *Distributed leadership:* each of the group members assumes leadership, encouraging both individuals and the group to make progress towards goals on both intellectual and managerial levels.
- *Mutual engagement:* group members co-construct, discuss/argue, or seek/offer help about mutually shared artifacts.
- *Peer Assessment and Feedback:* group members constructively evaluate *the results* of work done by themselves, their peers, and their group as a whole.
- *Group Reflection:* group members consider *the process* by which they will accomplish, are accomplishing, or have accomplished their tasks.

We see in our current research efforts [2] and ongoing experimentation that this system offers an environment in which L2L2 skills can be practiced in many scenarios. However, we recognize that presenting the learning environment without further support may not promote L2L2 explicitly, especially for novice learners, as other literature also suggests (e.g. [7],[8]). The challenge of promoting L2L2 explicitly necessitates identification of the key elements of social interaction. In this way, support and reflection can target these key elements to make collaborative learning effective.

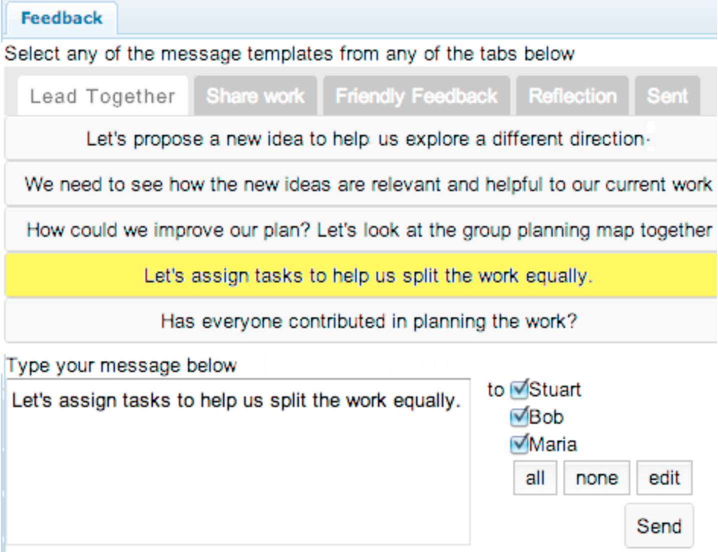
3 Promoting L2L2 through sending and recommending messages

As described earlier, our approach to L2L2 intervention and support is to provide a tool that guides and enables students to effectively interact with one another. Other research has demonstrated the potential benefits of supporting peer tutoring, (e.g. [9],[1]). Others are also taking the approach of using an AI system to recommend feedback that should be given by a human mentor [10]. We attempt to apply both of these principles to work within our L2L2 framework, where we encourage students to engage with peers by spontaneously taking on the role of mentor, providing timely feedback and initiate discussions about their learning process To enable and encourage students to engage in these activities, we developed a messaging tool that promotes students in using specific messages to engage in L2L2 and regulate their own collaboration. This tool provides students with the means to be their own facilitators, interacting with their peers or entire group as necessary. In addition, this same system provides a method for teachers and automated agents to offer similar interventions. In order to scaffold L2L2, the system offers specific speech acts, implemented as message templates, to focus students on the high-level concepts of L2L2. Creating well-targeted, supportive, and helpful message templates is crucial to the success of such an approach, and therefore we took an iterative, data-driven approach to understanding what specific speech acts might promote positive L2L2 behaviors. These speech acts, which were collected from actual student and teacher dialog, were then abstracted as message templates, applicable across the wide range of Metafora scenarios.

3.1 Sending and receiving messages

The Messaging Tool was developed to satisfy requirements that both our previous research with similar tools [11] and early pilots allowed us to identify. While providing some scaffolding for the previously mentioned reasons, the tool also had to be simple and speed up (rather than delay) interaction between students. In addition, we wanted to provide not only opportunities for reflection but also flexibility to students and the ability to adapt the messages to their specific situation and task. As such, the tool is equipped with what we refer to as *message templates,* sentences that correspond to the four L2L2 aspects and refer in a general manner both to the stages of the students' current activity, and the different tools they may be using (particularly the planning and the discussion tool).

Any group member can select one of these message templates and then potentially edit the template to adapt to the particular situation. The messages that are sent with the tool are kept for further reflection (Fig. 1, the "sent" tab). A snapshot of the tool appears below. Fig. 1 shows the tool from which messages are sent, and Fig. 2 demonstrates how the message appears for students receiving the message.



Feedback

Select any of the message templates from any of the tabs below

Let's propose a new idea to help us explore a different direction.

We need to see how the new ideas are relevant and helpful to our current work

How could we improve our plan? Let's look at the group planning map together

Let's assign tasks to help us split the work equally.

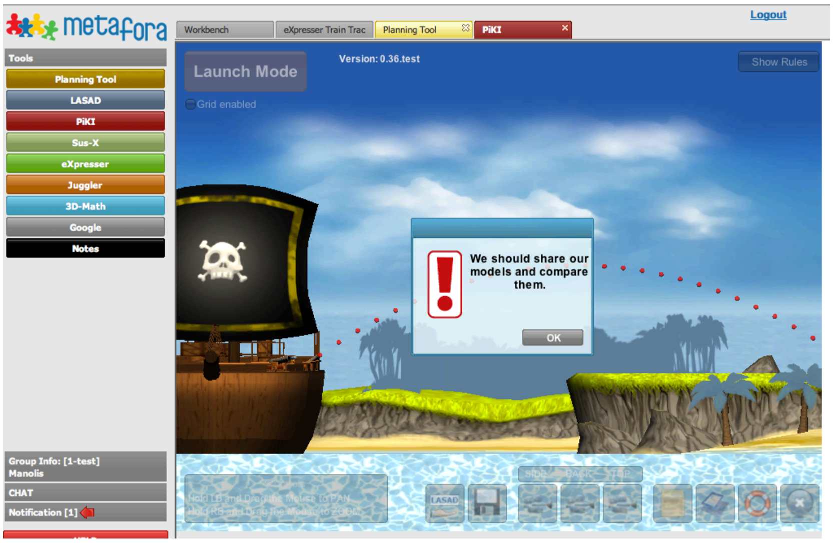
Has everyone contributed in planning the work?

Type your message below

Let's assign tasks to help us split the work equally.

to ☑Stuart
☑Bob
☑Maria

Fig. 1. The **Messaging Tool**. Students can choose and edit messages templates from each tab representing the different L2L2 aspects (the titles are adapted to children-friendly version)



We should share our models and compare them.

Fig. 2. Once a message is sent, it appears as pop-up anywhere that the students are working. In this case, a student is investigating their PIKI construction without much attention to the work of the rest of the group, and another student requests that they share and compare their work.

The system includes two types of message templates — peer and external — both created based on previous studies and Wizard-of-Oz experiments (c.f. [12]). Peer message templates are designed to address the group of students working together, and are sent by individual students to the rest of this group. These messages are designed to scaffold group work. External messages are equivalent messages that the system can send (whenever appropriate) as interventions. This list of 'external' messages can also be used by a teacher or any facilitators, who can launch the system separately and use it to support the students, as described in [12] where we presented similar work using these tools to simulate the provision of messages). Table A.2 in the appendix presents a tentative sample list of message templates.

3.2 Delivering and Recommending Messages

In early experimentation we observed a potential limitation of the messaging tool, in that it was challenging to identify quickly the most relevant L2L2 aspect and message templates. Taking into account that reflection is better encouraged when in context, we designed the system for highlighting (recommending) pertinent messages based on students' recent work.

This recommendation relies on a cross-tool analysis component that gathers historical data and can analyze pieces of evidence which we refer to as *indicators* (a statement of user activity from any tool in Metafora) or *landmarks* (a high-level statement of some abstract concept occurring in Metafora, indicative of accomplishment or need for remediation) that are generated by the different tools (for early steps in this approach see [13]).

Our challenge was to identify high-level student behaviors that call for intervention. From the superset of all L2L2 behaviors identified through data analysis, we select behaviors that are high-level enough to be directly relevant to L2L2 through conceptual links with the L2L2 definition, but also low-level enough to be directly mapped to certain actions within the system. Obviously, generality is a challenge, as each tool reports indicators and landmarks that are meaningful to the use of the specific tool, but not necessarily to the use of tools more generally. Therefore, we also require landmarks that can be understood in a generic sense across all tools, landmarks about which the cross-tool analysis component can reason. We have defined three broad labels for landmarks coming from the different tools that allow for cross-tool recognition and decision-making:

- *Perceived Solution*: an evaluation of an artifact produced within a tool that the students may consider a solution (but is not necessarily a solution).
- *Possible Solution*: a positive evaluation of the student's work that (based on some heuristics or criteria) is considered an acceptable solution to the given task.

- *Apparent Struggle*: some negative observation of a production process, outcome, or interaction that indicates intervention is necessary.

The cross-tool analysis component can then use these labeled landmarks and, in combination with the low-level action indicators, look for patterns across students that are indicative of L2L2 and provide opportunity for potentially fruitful intervention.

There are two distinct interventions that the automated support can send. First, a *direct message* exploits the system's interface for messages to directly present an L2L2 message (selected from the templates) to the student(s). This is a traditional form of AIED feedback, where students receive some targeted advice about their work from an automated system. This type of intervention has the advantage of directly requiring the students' attention, which can ensure students are receiving the necessary feedback. However, the direct approach has the disadvantages of being forceful and of taking control away from students.

In contrast, the second intervention method comes in the form of a *recommended message template*, a type of intervention where certain message templates in the messaging tool are highlighted in order to make clear which messages are most pertinent to the student's current situation.

We hypothesize that this recommendation intervention has multiple benefits. It has the potential to increase the students' involvement in the meta-level regulation of their own learning process, because the recommendations only hint to a student what might be most relevant, but still leave the onus on the student to engage in the L2L2 process. Additionally, a practical advantage to the recommendations is that if the AI system misjudges a situation, this will generally cause less harm. Table 1 contains examples of interventions as an outcome of analysis information shared by the tools for particular behaviors.

Table 1. Examples of mapping L2L2 behaviors to a specific pattern of indicators and landmarks that can be recognized by the cross-tool analysis component, which in turn can enact the given intervention. Examples of behaviors are related to the examples from section 2.2.

	Behavior	Indicators and Landmarks	Intervention
Distributed Leadership	Different members of the group should take the initiative to introduce and discuss new ideas.	- One person in the group creates a new resource. - Lack of discussion (in LASAD or chat).	*Recommended Message:* "This is a new idea. We should discuss how it is relevant and how it can help us."
utual Engage	Group should work together	- Divergence without convergence in plan-	*Recommended Message:* "Lets discuss why we have

	in a supportive and integrated way.	ning /reflection tool (Apparent struggle). - Lack of discussion (in LASAD or chat).	disagreed in LASAD, explaining first what is tricky about the task and what we are not so sure about."
Peer feedback and assessment	Group members should consider solutions offered by others and how those solutions relate to their own solutions.	-Apparent solutions from team members on separate computers -Apparent solutions not shared in LASAD, not accessed by other members	*Recommended Message:* "Lets evaluate one another's solution with respect to our task" *Direct Message:* "You should consider your solutions with respect to the task."
Group reflection	Group should re-visit and reflect upon their plan as they work	-Lack of plan revision with abundance of indicators from other tools. -Lack of attitude or Role cards	*Recommended Message:* "Let's revise the plan to show how we are going to work as a team." *Direct Message:* "You should consider how attitudes have played into your planning."

It is important to note the varied use of recommended messages vs. direct messages in the intervention column of Table 1. While each specific decision to send a direct message vs. recommendation can be debated from an instructional perspective, it is clear that certain situations may call for direct intervention because the situation is deemed as critical and the system has high confidence in its diagnosis. The difference between direct messages and recommended messages can also potentially be used as scaffolding, and faded over time. More direct messages early on can help students learn how and when these messages might be appropriate, and over time they can then be given only as recommendations, when students are expected to offer messages to one another in productive ways on their own.

Lastly, while this research is not focused on the teacher, this messaging system invites teacher participation as well, allowing them to send messages to student groups. Similarly, teachers can receive the recommendations from the system to help them quickly and easily identify the types of messages that are most likely necessary for any given group at a particular point in time. In this way, a single intervention system based on messages is acting as: 1) an intermediary for students to interact with each other, 2) a tool for teachers to interact with the students, and 3) a system for automated agents to offer intervention on varying levels of interruption.

4 Conclusion

This article presents an attempt to support social regulation in a collaborative environment known as Metafora with an explicit aim to support Learning to Learn Together (L2L2). The system, through both its design and automated support system, helps students become aware of many requirements of effectively learning with others in a group by explicitly referencing and drawing attention to the four L2L2 aspects. Since the Metafora platform and pedagogy are aimed at not only teaching domain knowledge — where approaches in AIED and ITS have demonstrated their potential — but also attempting to help students reflect on L2L2 by encouraging them to plan and regulate their own learning, we recognize that developing a 'traditional' intelligent system that sends feedback directly to students is not necessarily an adequate solution. Apart form the typical challenge of deciding when and how to provide feedback, there are conceptual challenges to ensuring this feedback encourages high-level reflection on L2L2 and that the feedback is generically available and applicable for all domains and learning scenarios.

This paper offers a new conceptualization of what an AI intervention (in the general sense) can look like: a system where fundamentally equivalent, theoretically grounded message templates can be utilized by different stakeholders (human or AI agent) according to the needs, abilities, and circumstances of the given scenario. Apart from making these message templates available for students to consider and exchange, the same basic messages can either be catered to be sent directly to students (with appropriate justification) or be recommended to students or teachers as potentially pertinent to the situation. Pilot experimentation suggests that these recommendations act not only as a practical means of helping students select from a large list of potential messages but also as a scaffold in suitable moments, to help students develop "L2L2" ways of thinking that can support them in becoming better group learners.

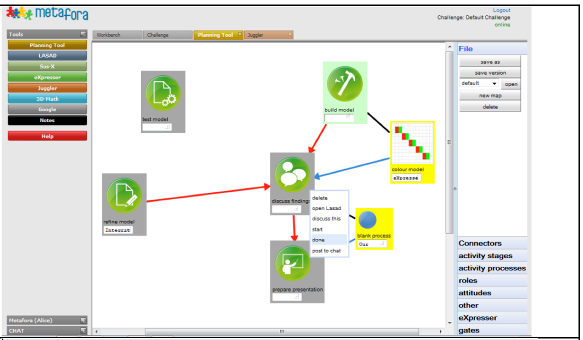
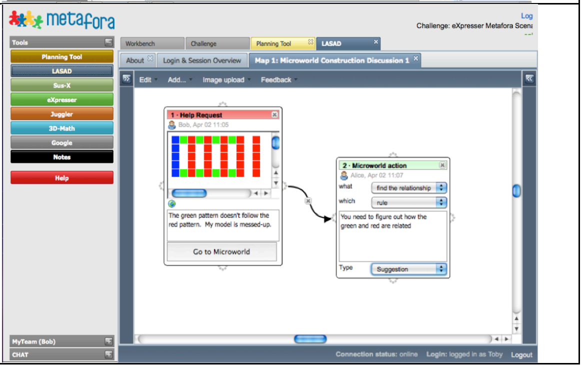
In future work, we intend to investigate in more detail the potential of both the availability of those messages in comparison with a less scaffold approach, and particularly the added value of the recommended messages vs. simply encouraging students to use the messaging system in general. Our hypothesis is that the sheer availability of the messages stimulates reflection and has the potential to improve awareness on L2L2. However, our previous work and initial pilots suggest that when messages are recommended based on relevance to the context, we will see even more significant behavioral changes in groups due to these messages, especially when students have ownership of the messages.

References

1. Walker, E., Rummel, N., Koedinger, K. R: Beyond explicit feedback: new directions in adaptive collaborative learning support. In Proceedings of the 9th international conference on Computer Supported Collaborative Learning - Volume 1 pp. 552-556 International Society of the Learning Sciences (2009)
2. Dragon, T., Mavrikis, M. McLaren, B.M., Harrer, A., Kynigos, C., Wegerif, R., Yang, Y.: Metafora: A web-based platform for learning to learn together in science and mathematics. IEEE Transactions on Learning Technologies. 10 Jan. 2013. IEEE Computer Society Digital Library http://doi.ieeecomputersociety.org/10.1109/TLT.2013.4 (2013).
3. Scheuer, O., McLaren, B. M.: CASE: A configurable argumentation support engine. IEEE Transactions on Learning Technologies, IEEE computer Society Digital Library. (2013)
4. Gokhale, A. A.: Collaborative Learning Enhances Critical Thinking. Journal of Technology Education, 7(1) 22-30 (1995)
5. Soller, A., Monés, A. M., Jermann, P., Mühlenbrock, M.: From Mirroring to Guiding: A Review of State of the Art Technology for Supporting Collaborative Learning. International Journal on Artificial Intelligence in Education, 15(4), 261-290 (2005)
6. Wegerif, R. Dialogic: Education for the Internet Age. Routledge (2013)
7. Dekker, R., Elshout-Mohr, M., & Wood, T.: How Children Regulate their Own Collaborative Learning. Educational Studies in Mathematics, 62(1), 57–79 (2006)
8. Dillenbourg, P., Jermann, P.: Designing integrative scripts. In Scripting computer-supported collaborative learning pp. 275–301. (2007)
9. Chaudhuri, S., Kumar, R., Howley, I., Rosé, C. P.: Engaging collaborative learners with helping agents. In Proceedings of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling pp. 365-372. IOSPress (2009)
10. Keshtkar, F., Morgan, B., Graesser, A.: Introducing an Intelligent Virtual Suggester and Assessment in Computer Educational Games. In the 11th International Conference on Intelligent Tutoring Systems (ITS-2012), Workshop on Intelligent Support for Learning in Groups 2012. Chania, Greece (2012)
11. Porayska-Pomsta, K., Mavrikis, M., Pain, H. Diagnosing and acting on student affect: the tutor's perspective. In User Modeling and User-Adapted Interaction, 18(1):125—173 (2008)
12. Mavrikis, M., Dragon, T., McLaren, B.M. Proceedings of the Workshop on Intelligent Support for Exploratory Environments 2012: Exploring, Collaborating and Learning Together at the 11th International Conference on Intelligent Tutoring Systems, ITS 2012. Chania, Greece (2012)
13. Dragon, T., McLaren, B., Mavrikis, M., Geraniou, E. Scaffolding collaborative learning opportunities: integrating microworld use and argumentation. Advances in User Modeling, 18-30 (2012)

Appendix

Table A.1. Tools used in all learning scenarios

Planning/Reflection Tool: provides a visual language to support students in planning and reflecting; activities, roles, resources, task assignments, and attitudes are visualized, discussed, and reflected upon.	
Discussion Tools: provide a shared workspace for students to have in-the-moment chat, as well as structured discussions and argumentation, through a graphical argumentation tool, LASAD (see more info https://cscwlab.in.tu-clausthal.de/lasad/)	

The green pattern doesn't follow the red pattern. My model is messed-up.

what: find the relationship

which: rule

You need to figure out how the green and red are related

Type: Suggestion

Table A.2. Examples of message templates to be sent by students or to be recommended by the system. Note that each message also has an equivalent message with adapted language and grammar that appear as external to the group and can be used from the system as a direct message. For example, instead of "Let's look…" "Everyone should look…"

	Message Template	Comments
Distributed Leadership	Let's propose a new idea to help us explore a different direction.	Useful in a phase of brainstorming as a means of getting the team out of an impasse.
	We need to see how the new ideas are relevant and helpful to our current work.	Highlights the importance of regulatory moves during idea generation and provides an example of criteria for accepting or rejecting ideas.
	Let's look at the group planning map together.	Relevant when some students' activities seem to be diverting from the plan.
	How could we improve our plan?	Inspires specific leadership moves from members of the team. These messages promote the equal share of both work and leadership
	Let's assign tasks to	

	help us split the work equally.	(planning) from all the members of the team.
	Has everyone contributed to planning the work?	
Mutual Engagement	Has everyone done the work they said they would do?	Similar to the last two messages of the previous category, but intended to refer to engaging particularly with the discussion or work in the microworlds.
	Has everyone contributed to the discussion?	
	I/We need some help with <…>	Promotes peer help-seeking --- students are often reluctant to ask for help from peers even when stuck.
	We seem to disagree. Have we all understood each other's opinions?	Helps students step back from the "heat of the disagreement" and fosters shared understanding and by encouraging students to rethink the problem and help reach consensus and/or generate new action.
	Lets discuss our conflict starting from the causes of our confusion.	
	We seem to disagree. Lets redefine our group goals/attitudes/roles.	Defining goals/attitudes or roles involves students in a discussion about their different perspectives.
Peer feedback and assessment	We should share our models and compare them.	Sharing and comparing models promotes meaning-making with respect to the domain.
	Lets evaluate one another's solution with respect to the task.	Constructive peer assessment is an important skill but students often ignore the original task and tend to focus only on procedural rather than conceptual aspects hence this message recommends specific criteria.
	Let's explain clearly in our evaluation what is the problem	
	Let's revise our plan. Does it match our work so far?	Revising the plan at specific phases during and at the end of the collaborative process initiates reflective discussions.
	Let's use the attitude/role cards to reflect on our work so far.	Employing attitudes and roles in the plan encourages reflection on the collaborative process at the meta-level.
	Lets consider our best/worse moment as team so far.	A message often used in critical incident analysis as a way of reflecting and generating meaning out of events.

Supporting Collaborative Learning in Virtual Worlds by Intelligent Pedagogical Agents: Approach and Perspectives

Mohamed Soliman

IICM, Graz University of Technology, Graz, Austria
muhamed.soliman@gmail.com

Abstract. Intelligent pedagogical agents (IPA) are aimed to support learning in virtual worlds. Motivations for adopting IPAs in virtual worlds are to compensate for lack of human pedagogical presence, to improve student engagement, and having autonomous support. Given named challenges to realizing IPAs in virtual worlds, a proposed solution approach is to simulate IPAs with targeted scenarios with intelligent agents prior to realization. This paper discusses intelligent agent based simulation of a collaborative learning scenario that facilitates IPA support to collaborative learning in virtual world. The collaborative learning scenario is composed of multiple avatars interacting to conduct an experiment simulation in a virtual world with an IPA. The paper discusses types of support the agent will do to scaffold the interactive collaborative learning activity, for example by mediating interaction among learners and targeting learning to collaborate as well as collaborating to learn with benefits shown.

Keywords: CSCL, Intelligent Pedagogical Agents, Intelligent Agents

1 Requirements

A collaborative learning activity design is motivated by the objective to employ Intelligent Pedagogical Agents (IPAs) in virtual worlds to support learning. While there are different means to support collaborative learning in virtual worlds (Dalgarno, 2010), automated and artificially intelligent pedagogical support are still needed. Design objectives of IPAs are to provide automated and intelligent pedagogical support while improving engagement throughout interactivity. While there are different roles the IPA can do to support collaborative learning activities in a virtual world, there is the importance of focusing on interaction among learners and with a leaning object in relation to situated learning and learning by doing. Prior works (Soliman & Guetl, 2010; Soliman & Guetl, 2013) highlighted other possibilities of IPA support.

In contrary to an individual learning scenario, the IPA role has to shift towards being more of a mediator that facilitates the dialogue and interaction among the learners and the learning object in the collaborative setting. An important task of the IPA is to maintain distribution of roles, as a key component, among different learners (Hoadley, 2010). Distribution of roles in the task is assumed to be available as an input to the learning activity. The IPA is assumed to be executing a micro level script rather than a macro level to discover details of interaction as a design objective (Kollar, Fischer, & Hesse, 2006; Weinberger, 2011). Selection of the group size is determined to start with two learners agreeing to what is cited by Hoadley (2010), *"Stahl (2006) has argued that the small group level is the 'sweet spot' for studying CSCL"*.

The targeted scenario is described by two avatars performing an experiment simulation with the aid of an IPA. The avatars are human controlled while the IPA is an autonomous agent. The IPA supports the learning activity with the following:

1. Provide tutorial about the experiment. In collaborative learning scenarios, the IPA will intervene only to scaffold learning after giving the opportunity to other learners to learn to collaborate.
2. Providing motivational support.
3. Answer questions. In the group learning, the IPA will rather stimulate group interaction before answering a question individually.
4. Support the collaborative activity such as "who is supposed to perform this task?"
5. Promote reflection and trans-activity (Boud, Keogh, & Walker, 1985) as important components to collaborative experiential learning.
6. Provide varying levels of support from the learner level to the group level.
7. Ensure continuation of the activity, to manage idle time behavior for example.

However, several challenges exist for implementing an IPA directly into the virtual world, Soliman and Guetl (2013). Hence, simulating the collaborative learning activity in the intelligent agent framework is useful. This is to focus on interactivity and intelligence support to the collaborative learning activity and to identify how an intelligent agent can complement the IPA functions in particular to the collaborative interaction.

2 Solution Approach

2.1 BDI-Based Collaborative Learning Scenario Simulation

The BDI agent framework of Jadex (Jadex, 2013) is adopted as a result of evaluation and selection steps (Soliman & Guetl, 2012). Inter-agent communication is used to simulate the players' interaction in the learning activity communication towards enabling its analysis and reasoning. In BDI based environments, multi-agent design involves determination of goals, plans, events (or messages), and beliefs. Goals represent static or dynamic desires the agent should pursue, plans represent intentions (as recopies of the solution) translating into actions. Beliefs represent agent knowledge about the environment and other learners and can also change dynamically according to events. A BDI based collaborative learning scenario simulation involves determination of goals, plans, and beliefs.

2.2 Settings and Design

Setting the experiment implies simulating the players (actors and artifacts) of the scenario in the virtual world. Four agents are defined: an agent representing the IPA, two agents representing the learner avatars, and an agent that simulates the intelligent object (device) behavior in the virtual world. The BDI-based agent design requires setting the beliefs, desires, and intentions of the agents:

- *The IPA agent* has beliefs about learners, the task, and the roles. The desire of the IPA is a pedagogical goal to facilitate (direct) the completion of activity. The intentions of the IPA are plans representing variations according to interactions.
- *The device agent* represents an experiment. It gives an autonomous behavior property to the object to simulate different results that can be handled in learning settings by learners or the IPA.
- *Two learner agents* are allocated. The desire of each learner agent is to accomplish the learning experiment in collaboration with another learner. Intentions adopt sequences in results to interaction. Beliefs add details of the learner knowledge about the other learner.

2.3 Interaction & Collaboration

The IPA initiates the first step to run the experiment and finds, in the role-responsibility beliefs, which learner is allocated issuing a request for the assigned learner agent to start. If the correct action is performed, it updates the assessment belief base. If the task is wrong, as observed from the device, the IPA records and triggers collaborative discussion with the other learner. The task is repeated (according to pre-set number of trials) by the same learner (if a capable learner can show the task, it can be performed by another learner). Otherwise, the IPA can give a demonstration of how the task if performed and move to the next task. The IPA will continuously monitor the interaction identifying which agent is responding. Consecutive tasks will proceed until the experiment completes. Before each step, IPA sends a message to both learners to trigger discussion on how to perform the next task. In each step, if the wrong learner responds, IPA issues an error message while recording the result into the assessment belief set. Directing messages to both learner agents serves the learning to collaborate objective. Furthermore, when the IPA recognizes long idle time, it asks both learners to discuss roles and the expected task on which action to take.

3 Concluding Remarks

The learning scenario is implemented in Jadex as a selected agent platform to avoid difficulties of actual implementation. The simulation of this scenario in the agent based environment helps to:

1. Isolate implementation difficulties in a virtual world.

2. Discover means of IPA support for collaborative scenario – how the collaboration scenarios will take place in a virtual world implementation.
3. Discover means of interaction design for the learning scenario in relation to roles.
4. Requirements from the learning object to support the learning interaction from one learner in relation to more than one learner.
5. Investigations into integrating micro-level collaborative scripts and contributing a collaborative pattern of IPA in virtual world based learning.

References

Boud, D., Keogh, R., & Walker, D. (1985). Introduction: What is Reflection in Learning. In Boud, D. et al. (Eds.), Reflection: Turning experience into learning. London: Kogan Page.

Bratman, M. Intention, Plans, and Practical Reason. (1987). Harvard University Press. Cambridge, MA, USA.

Dalgarno, B., & and Lee, M. (2010). What are the Learning Affordances of 3-D Virtual Environments? British Journal of Educational Technology. Vol 41 No 1.

Gaillet, L. (1994). An Historical Perspective on Collaborative Learning. Journal of Advanced Composition, 14 (1).

Hoadley, C. (2010). Roles, Design, and the Nature of CSCL. Journal of Computers in Human Behavior. Vol 26, Issue 4 (pp. 551-555).

Jadex, (2013). http://jadex-agents.informatik.uni-hamburg.de, Accessed May 31, 2013.

Jaques, P., Andrade, A., Jung, J., Bordini, R., & Vicari, R. (2002). Using Pedagogical Agents to Support Collaborative Distance Learning. CSCL'02, Boulder, Colorado.

Kobbe, L., Weinberger, A., Dillenbourg, P., Harrer, A., Hämäläinen, R., & Fischer, F. (2007). Specifying Computer-supported Collaboration Scripts. International Journal of Computer-SupportedCollaborative Learning, 2(23), 211–24.

Kollar, I., Fischer, F. & Hesse, F. W. (2006). Collaboration Scripts – a Conceptual Analysis. Educational Psychology Review, 18(2), 159–85.

Schmeil, A., & Eppler, M. (2009). Knowledge Sharing and Collaborative Learning in Second Life: A Classification of Virtual 3D Group Interaction Scripts. International Journal of Universal Computer Science (JUCS), 15(1).

Soliman, M., & Guetl, C. (2013). Implementing Intelligent Pedagogical Agents in Virtual Worlds: Tutoring Natural Science Experiments in Open Wonderland. The IEEE Global Education Conference, IEEE EDUCON 2013, March 12-15 2013, Berlin, Germany.

Soliman, M., & Guetl, C. (2012). Experiences with BDI-based Design and Implementation of Intelligent Pedagogical Agents. International Conference on Interactive Computer-Aided Learning, ICL2012, Sept. 2012, Villach, Austria.

Soliman, M., & Guetl, C. (2010). Review and Perspectives on Intelligent Multi-agent Systems' Support for Group Learning. World Conference on Educational Multimedia, Hypermedia & Telecommunications ED-MEDIA 2010, June 2010, Toronto, Canada.

Stahl, G., Koshmann, T., Suthers, D. (2006). Computer Supported Collaborative Learning: a Historical Perspective. In Sawyer (Ed.), Cambridge handbook of the learning sciences, Cambridge University Press.

Vygotsky, L. (1978). Mind in society. The development of higher psychological processes. Cambridge: Harvard University Press.

Weinberger, A. (2011). Principles of Transactive Computer-Supported Collaboration Scripts. Nordic Journal of Digital Literacy. No. 03.

Towards semantic descriptions of collaboration indicators to support collaboration models transferability

Jesús L. Lobo[1], Olga C. Santos[2] and Jesus G. Boticario[2]

[1] Technology & Society, Tecnalia Research & Innovation, Basque Country, Spain
[2] aDeNu Research Group. Artificial Intelligence Dept. Computer Science Sch. UNED. Spain
[1]jesus.lopez@tecnalia.com, [2]{ocsantos,jgb}@dia.uned.es

Keywords: Collaborative learning, Transferable collaboration model, Collaborative indicators, Collaboration analysis

1 Research

Students around the world are currently taking advantage of e-learning platforms to support their learning, and one of the most important features in some of these platforms is their support for collaborative learning. In this context, a collaboration analysis is necessary to ascertain whether collaboration takes place. Having this in mind, data mining techniques are often used to identify student collaboration indicators based on their forum interactions (see relevant literature elsewhere).

The Collaborative Logical Framework (CLF) system, based on an approach used by international Cooperation Agencies, sets guidelines to promote participation in CSCL [1]. It is fully integrated into dotLRN/OpenACS as one of its packages and consists of making the students work consecutively in three ways: 1) solving tasks individually 2) working in cooperation with their colleagues' to improve own solutions, and 3) working all together to reach an agreement for the joint solution. Moreover, the system gathers the students' performance to infer how they work in the course. By means of a varied number of metrics, derived from the analysis of forum interactions, the system provides their behavior related to the collective task. In particular, these metrics focus on ratings given to their colleagues' contributions, on the revised versions they create of their solutions after the colleagues feedback received, and studying the actions they carry out before and after a specific interaction. This information helps the student and the tutor to monitor the tasks, and on the other it is used to get collaborative indicators, which define the learner's reputation.

Domain-independent statistical indicators of students' interactions in forums (conversations started, messages sent, and replies to student interactions) were identified elsewhere by mining non-scripted interactions in dotLRN and evaluated the benefits of their awareness by students [2]. In this context, the objective of this work is to enrich student's meta-cognitive support in the CLF by adding these automatically inferred and validated indicators (focused on initiative, activity and regularity, and perceived reputation) using the CLF metrics to express them.

If possible, our intention is to use available standards and specifications to semantically model the indicators and support transferability of collaboration models among different systems.

Besides well-known benefits of collaboration awareness in motivating students' collaboration, indicators inferred can be also used to provide adaptive features to the e-learning system. Thus, depending on the student collaboration profile and behavior, the system can react accordingly by providing individual suggestions. The goal here is to identify recommendation opportunities that guide the student to perform specific actions in order to help on the task, encourage participation and improve team work.

2 Suggested Topics for Discussion

- Descriptions of collaboration indicators modeling in terms of available standards to support transferability of collaboration models among systems.
- Elicitation of recommendation opportunities to manage and guide collaboration.

3 Biography

Mr. Jesús L. Lobo has an MSc in Computer Engineering (Deusto University, Spain, 2003) and he is working at Tecnalia Research & Innovation as Projects Responsible and ICT Consultant. His work is mainly focused on key activities such as e-skills, technology for learning, ICT certification processes, and ICT and lifelong learning activities.

Dr. Olga C. Santos is aDeNu's R&D Technical Manager and has contributed to 14 projects and over 150 papers and 50 scientific committees researching on affective inclusive personalized adaptive navigation support in ubiquitous standards-based social online learning environments.

Dr. Jesus G. Boticario is professor at the Computer Science School and has held several positions at UNED. He has published over 200 research articles and participated in 24 R&D funded projects. Head of aDeNu research group, scientific coordinator in European and National projects and co-chairs of workshop series on User Modeling and Accessibility.

Relevant Publications

1. Santos, O.C., Rodríguez, A., Gaudioso, E., Boticario, J.G.: Helping the tutor to manage a collaborative task in a web-based learning environment. In: AIED2003 Supplementary Proceedings, 153-162 (2003)
2. Anaya, A.R., Boticario, J.G.: Content-free collaborative learning modeling using data mining. In: User Modeling and User-Adapted Interaction 21, 181-216 (2011)

Scripting at the tabletop to improve collaboration

Andrew Clayphan

School of Information Technologues, The University of Sydney, NSW, 2006, Australia
andrew.clayphan@sydney.edu.au

Keywords: Tabletops, Interactive Surfaces, CSCL, Scripting

1 Research

Interactive collaborative tabletops are promising devices that can help collocated people collaborate because they augment natural round-table discussions with a shared digital space that offers equal opportunities of actions and access to resources available. We propose collaborative scripts for enhancing tabletop collaboration in the form of: guidance and structure; advice on how to do the task; and control over constraints afforded by the tabletop.

After studying the ways people have used tabletop interfaces, we concluded that it is valuable to define scripts that will help people collaborate more effectively in co-located, technology-enhanced scenarios [3]. Different from scripts investigated so far, our work allows learners to negotiate over the scripts – initially explored in the domains of brainstorming, concept mapping, and collaborative poster creation.

Brainstorming – a technique to encourage creativity in small groups. Our method separates the technique into three stages: idea generation; idea organisation and reflection [1]. Each stage is scripted through the use of negotiation elements that alter a stage. The system presents a choice between users leading negotiation or a facilitator making choices, for example: whether to enable touch input; whether to colour ideas (to show authorship); etc.

Concept Mapping – a technique to help learners represent knowledge about a given topic in a graphical format, making use of meaningful propositions to link concepts in a domain of interest. Building a concept map at the tabletop can help students visualise different perspectives of the same topic and trigger discussions towards agreement on main ideas that describe the knowledge domain [4]. Collaborative scripts are set to drive groups of students to produce better quality concept maps, for example: the layout of concepts according to different theoretical principles.

Collaborative Poster Creation – designed for small groups to build a joint artefact from personal collections [2], consisting of an individual collection stage, and then collaborative stages of sharing and building. The collaborative stages have potential for scripting, for example: enforcing viewing of content – before being permitted to advance in the task.

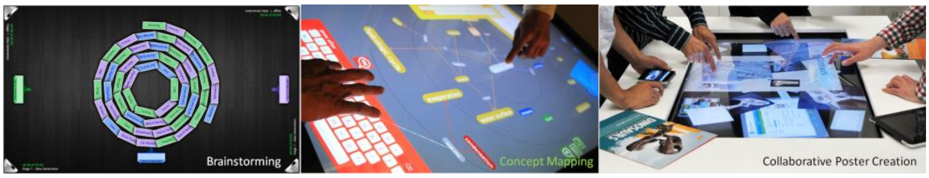


Figure 1. Examples of tabletop applications used for exploring scripting.

Each activity, presents design issues to consider when formulating a set of guidelines to consider for scripting at the tabletop. These are: (1) People have different expectations and knowledge of the task at hand. (2) Voting/negotiation mechanisms – the way a group resolves issues. (3) The need for sound default settings. (4) Identifying group collaboration and how to show this to learners. (5) Whether the main task was executed as expected, and the role scripting had towards this.

We propose a set of guidelines: (1) Regulate learning activities [6] – keep "*activities of learners coordinated and guided according to particular rules, implemented via respective tools in the learning environment*" [5]. (2) Foster collaboration – organise the activity and the script to promote collaboration. (3) Facilitate egalitarian participation. (4) Define level of user control. (5) Foster awareness – develop an understanding of other participant actions. (6) Adjust the script based on information from the system and the users. (7) Use Tabletop Affordances – take advantage of the constraints introduced by the tabletop, such as: face to face discussion; and methods to exploit the hardware.

2 Suggested Topics for Discussion

- Whether script approaches at the tabletop should be system or role based or both?
- The representation of open learner models to aid in the scripting process?
- The appropriate level of feedback for learners? OLM's?
- Methods to help determine if a script is needed?

3 Biography

Andrew Clayphan is a Ph.D. student at the CHAI Research Group at Sydney University, Australia. He holds degrees in Software Engineering (University Medal, Honours Class 1) and Finance from the University of New South Wales, Australia.

Relevant Publications

[1] A. Clayphan, J. Kay, and A. Weinberger. ScriptStorm: scripting to enhance tabletop brainstorming. In Personal and Ubiquitous Computing, (to appear), 2013.

[2] A. Collins, A. Clayphan, J. Kay, and J. Horder. My museum tour: Collaborative poster creation during school museum visits. In EIST2012.

[3] L. Kobbe, A. Weinberger, P. Dillenbourg, A. Harrer, R. Hämäläinen, P. Häkkinen, and F. Fischer. Specifying computer-supported collaboration scripts. In IJCSCL 2(2):211–224, 2007.

[4] R. Martinez, J. Kay, and K. Yacef. Collaborative concept mapping at the tabletop. In ITS2010, pages 207–210, 2010.

[5] A. Meier, H. Spada, and N. Rummel. A rating scheme for assessing the quality of computer-supported collaboration processes. In IJCSCL, 2(1):63–86, 2007.

[6] A. Weinberger. Principles of Transactive Computer-Supported Collaboration Scripts. Nordic Journal of Digital Literacy, 03:189–202, 2011.

AIED 2013 Workshops Proceedings
Volume 4

AIED Workshop on Simulated Learners

Workshop Co-Chairs:

Gord McCalla
Department of Computer Science, University of Saskatchewan

John Champaign
RELATE, Massachusetts Institute of Technology

https://sites.google.com/site/aiedwsl/

Preface

This workshop is intended to bring together researchers who are interested in **simulated learners**, whatever their role in the design, development, deployment, or evaluation of learning systems. Its novel aspect is that it isn't just a workshop about pedagogical agents, but is also concerned about other roles for simulated learners in helping system designers, teachers, instructional designers, etc. As learning environments become increasingly complex and are used by growing numbers of learners (sometimes in the hundreds of thousands) and apply to a larger range of domains, the need for simulated learners (and simulation more generally) is compelling, not only to enhance these environments with artificial agents, but also to explore design issues using simulation that would be otherwise too expensive, too time consuming, or even impossible using human subjects. The workshop aims to be broadly integrative across all possible roles for simulated learners.

July, 2013
Gord McCalla & John Champaign.

Program Committee

Co-Chair: Gord McCalla, *University of Saskatchewan* (mccalla@cs.usask.ca)
Co-Chair: John Champaign, *Massachusetts Institute of Technology* (jchampai@mit.edu)

Esma Aimeur, *Université de Montréal*
Roger Azvedo, *McGill University*
Gautam Biswas, *Vanderbilt University*
Tak-Wai Chan, *National Central University of Taiwan*
Robin Cohen, *University of Waterloo*
Ricardo Conejo, *Universidad de Málaga*
Evandro Costa, *Federal University of Alagoas*
Michel C. Desmarais, *Ecole Polytechnique de Montréal*
Sylvie Girard, *Arizona State University*
Lewis Johnson, *Alelo*
Yang Hee Kim, *Utah State University*
James Lester, *North Carolina State University*
Noboru Matsuda, *Carnegie Mellon University*
Kurt VanLehn, *Arizona State University*
Beverly Park Woolf, *University of Massachusetts, Amherst*

Table of Contents

Using Simulated Learners and Simulated Learning Environments within a Special Education Context

Carrie Demmans Epp[1] and Alexandra Makos[2]

[1] Technologies for Aging Gracefully Laboratory (TAGlab), Dept. of Computer Science
University of Toronto, Toronto, Canada
carrie@taglab.ca
[2] Ontario Institute for Studies in Education (OISE)
University of Toronto, Toronto, Canada
alexandra.makos@mail.utoronto.ca

Abstract. The needs of special education populations require specific support to scaffold learning. The design and use of intelligent tutoring systems (ITS) has the potential to meet these needs. Difficulty in the development of these systems lies in their validation due to the ethics associated in studying learners from this population as well as the difficulty associated with accessing members of this learner group. This paper explores the use of simulated learners as a potential avenue for validating ITS designed for a special education population. The needs of special education learners are discussed. Potential avenues for employing simulated learners and simulated learning environments to test ITS, instructional materials, and instructional methods are presented. Lastly, the expansion of an educational game designed to develop emotion recognition skills in children with autism spectrum disorder is used to illustrate how simulated learning environments can be used to support the learning of these students.

Keywords: Special Education, Ethics, Simulated Learners, Simulated Learning Environments

1 Introduction

Many intelligent learning environments have been shown to help learners who belong to the general population, but few existing systems have been shown to meet the needs of those who fall under the umbrella of special education [1]. Learners in this category have highly differentiated needs that are specified in an individual education plan (IEP) [2]. Their increased need for personalization and continuous reinforcement makes the argument for augmenting their education with intelligent tutoring systems (ITS) even stronger. However, this has not been done widely.

Several factors may contribute to the lack of ITS use within special education. The lack of validation that has been performed on the systems for special education populations [1], the difficulty of integrating ITS into special education settings [3], and the difficulty of designing activities that ensure deep understanding may contribute to the lack of ITS that support this population. The variability of learner needs presents additional challenges for system designers with respect to content development [3]. Furthermore, challenges that relate to the motivation, attitude, and social vulnerability of members of this population make it more difficult to design and validate systems. Developing systems for the special education population as a whole is difficult [4].

In addition to the above challenges, it may be difficult for designers to obtain access to a sufficiently large sample of the population to ensure that their ITS is beneficial in special education contexts. This is where the use of simulated learners and simulated learning environments can be advantageous since their use can mitigate the challenges presented by limited access to this vulnerable population and reduce the negative ethical implications of testing these systems on members of this population.

It is important to look at the research on situated learning in order to understand the achievements in best practices and lessons from research on simulated learning. Critical to this research is the combination of immersion and well-designed guidance that supports the situated understanding of learners whereby they not only have a deep understanding of the particular concepts that are being targeted, but the learners are able to then generalize and apply these learned concepts to other contexts [5]. Research shows that game-like learning through digital technologies is a viable tool across disciplines [6] and suggests that elements of game-like learning scaffold and guide learners towards a deep understanding of concepts. The on demand instruction of information that is vital to progress in the game is also important [5] and can be exploited to encourage learning. Simulations can include these elements and use stimuli to which special education populations react positively. Some stimuli that have been shown to increase student engagement include music, visual cues, and social stories [7]. Not only do these "strategies…help teachers increase engagement [but they] are vital for promoting positive outcomes for students" [7].

To support the argument for the use of simulated learners in this educational context, we first describe the characteristics and needs of this population as well as the learning environments in which they can be found. Following this, we discuss the use of ITS by special education students, which includes student interactions with agents. After laying this groundwork, we discuss the ethical implications and potential benefits to using simulated learners for validating ITS for use by special education populations. We then describe the potential uses of simulated learners and learning environments. This includes the description of an educational game, called EYEdentify, which was designed to develop emotion recognition skills in children with autism spectrum disorder (ASD). A discussion of how gaming principles and simulated environments can be further employed to expand EYEdentify for the purposes of helping scaffold learners' social interactions is provided.

2 Special Education

An introduction to the learning environments that exist in schools and the needs of learners who are classified as special education is presented. The use of agents and other forms of intelligent tutoring, within special education contexts, is then provided.

2.1 Learners and Learning Environments

These learners are either segregated into dedicated special education classrooms or integrated into classrooms whose majority population consists of learners from the general student body. Research has explored the design and integration of ubiquitous technology into special education classrooms [8], but few e-learning environments have been created to specifically support these students.

The needs and abilities of this population are highly variable, which can make generalizability hard [9]. This variability can be used to argue for the importance of personalizing students' learning materials, environments, and experiences, which is evidenced by the existence of IEP that detail the learner's specific needs and the accommodations that can be used to help the learner succeed [2]. Some of these accommodations include providing learners with additional time in order to complete tasks [1] or allowing learners to perform tasks using different modalities (e.g., oral responses rather than written ones) [2]. While these accommodations are necessary to ensuring the learner's success, it can be difficult to provide the necessary support, especially in integrated classrooms. The use of ITS that better support the individual needs of these learners could help alleviate the teacher's need to provide these supports.

2.2 Simulated Learner and Agent Use

While the use of agents within ITS used by special education populations has been studied, it appears that researchers and system developers are not simulating learners who have special needs. Nilsson and Pareto have instead used teachable agents within a special education context to help learners improve their math skills [3]. However, they experienced difficulty integrating the ITS into the classroom. Whereas, Woolf et al. were able to integrate their ITS into a classroom that had a mixed demographic: the class consisted of both low and high performing students, and of those who were low-performing, one third had a learning disability [10]. In this case, students interacted with an agent who played the role of a learning companion in order to support the learner's affective needs. It was found that this approach was especially beneficial to the low-performing students in the study, which may indicate the potential that this system holds for helping many of the learners who fall under the special education umbrella. Other work has also shown that interactions with agents within an ITS can improve or maintain learner interest and motivation [1].

3 Ethics

Given the vulnerable nature of this population, it is important that we not increase the risk that they are exposed to by introducing them to ITS or other learning techniques that have not been properly vetted since these could threaten the emotional well-being of learners or their learning success [11]. The use of simulated learners can help ensure that these systems are properly tested before we expose special education learners to them. Simulated learners can help teachers, instructional designers, and system developers meet the ethical guidelines of professional bodies by providing evidence of the limitations and appropriateness of the instructional methods used by systems or of the system itself [12].

4 Potential for Simulated Learner Use

We foresee two potential uses for simulated learners within a special education context both of which have been explored within other contexts. The first is during the development and testing of ITS [13, 14], and the second is for teacher training [13]. Using simulated learners in these ways provides developers and instructors with access to learners in this population and prevents any potential harm that could result from experimenting with members of this population. However, it may create a false sense of the validity and usefulness of different systems and instructional techniques, especially when we lack a full understanding of the abilities and symptomology of some members of this population (e.g., those with Phelan-McDermid Syndrome).

Generalizability is difficult to perform with this population [9], but some level of generalizability is required if a system is to be used by many people. Unfortunately, current design methods, such as participatory design, fail to address how the system's use and design should change over time. Furthermore, most users are unable to predict how they will use a system until they have integrated that system into their environment [15]. Carrying these challenges into the special education domain increases their severity because of the additional communication barriers that may exist between system designers and learners with special needs [4]. While observation is a component of many design methods, the lack of access to this population when combined with the communication challenges that exist reduces the feasibility of employing many of the more traditional user-centered design techniques.

Using simulated learners could benefit system designers and developers by allowing them to evaluate a system with various members of the special education population. This could reduce demands on a vulnerable population while allowing for some level of system validation to be performed. Furthermore, the use of simulated learners would allow systems to be tested with a far greater variety of learner types in order to identify where the system may or may not be beneficial. If the system were web-based, the simulated learners could be implemented using a Selenium test suite based on behavioural models of the system's target learners.

To effectively use simulated learners in this context, it is important to create these learners using different and competing theoretical models of their behaviours and abilities. This also alleviates some of the concerns that have been expressed over the use of simulated users when testing adaptive systems [16]. The source of these models can be teachers or special education experts since their mental models might inform good stereotype-based models of learners that capture general behaviours which are grounded in the expert's classroom experience. For example, haptic feedback can be used to reinforce certain behaviours (e.g., pressing a button) in children with ASD.

However, we would argue for also including models from other sources since the above experts are in short supply and cannot provide sufficient diversity in the models to ensure that systems are adequately tested for a general special education population. Simulated learners can be created from the cognitive models that are currently described in the educational psychology literature or through the application of educational data mining and learning analytics techniques to the logs of ITS usage where low performing and special education students were included in the classroom intervention. An example from the educational psychology literature could consider models of attention deficit hyperactivity disorder (ADHD), which include the amount of hyperactivity and inattention that a learner has, to create simulated students that behave in a way that is consistent with both the inattention that is known to affect individual outcomes and the hyperactivity that can affect the classroom environment for all students. Thus, allowing teachers to explore strategies that minimize the impact of both of the behaviours that characterize students with ADHD [17].

The diversity of models on which the simulated learners are based may help compensate for the inaccuracies that are inherent to modeling techniques, therefore, reducing the need for simulated learners to have high-fidelity cognitive models. Especially, since there is an incomplete understanding of the cognitive processes of all those who fall under the umbrella of special education, as is demonstrated by research in mathematics and learning disabilities [18].

That said, simulated learners that are based on these models could be used to validate the design of learning materials and to ensure their effectiveness or comprehension [13, 14]. Teachers could use simulated learners to test learning materials for their ability to increase learner engagement across a variety of contexts [7] before trying the materials on learners in their class. This would give teachers the opportunity to refine their teaching materials and confirm their suitability for students in the class.

Simulated learners can also be used to help prepare teachers either during preservice training or before a new school year begins when the teacher is preparing for his/her incoming students [13]. The use of agents who play different types of special education learners reduces the need to worry about the possible negative consequences that mistakes would have on learners [19]. This use of simulated learners also holds the potential to reduce teacher errors since teachers can try new techniques with the simulated learners and learn from those experiences, which may reduce the risk of their committing errors with live learners.

5 Potential for Simulated Learning Environment Use

While simulated learning environments can pose a threat to learning because of the complexity of the learning experience [20], they still hold the potential to benefit learners with special needs. Simulated environments allow learners to take risks in order to develop a deeper understanding of the situations they encounter [5]. This can increase learner awareness of potential situations that could be encountered when interacting with others. Ideally, simulated learning environments would be used to help the learner develop and transfer skills into the real world by gradually increasing the external validity of the tasks being performed.

Simulations allow system designers to ensure that the problems or activities being studied resemble those that learners experience outside of the simulation [1] and they allow for the gradual increase in the complexity and ecological validity of tasks [21]. This means that learners can begin their learning activities in a simpler environment that is safe and progress towards more realistic situations, enabling the use of van Dam's spiral approach, where learners encounter a topic multiple times at increasing levels of sophistication [22]. This can help learners transfer their developing skills into the real world. Additionally, the use of simulations accessible on different technologies can shift learner dependence on experts to technology whereby learner use of the technology can help learners gain a sense of independence and begin to develop the skills required to expand and extend their interactions to the real world [23]. We illustrate this trajectory through a discussion of a mobile game that was designed to help children with autism spectrum disorder learn to recognize emotions.

5.1 EYEdentify: An Educational Game for Emotion Recognition

EYEdentify is a mobile application for the Android platform that is designed to develop the emotion recognition skills of children with ASD since these are lacking. Previous technologies that have tried to teach this skill to children with ASD have primarily focused on the use of videos to model emotions for the learner [24]. Current research focuses on social skill development through the use of interventions that use a video series to develop social skills by exploiting the relationship between facial expressions and emotion [4, 25]. Emotion recognition research suggests the most important features of the face necessary to correctly identify emotions are the eyes and the mouth [26]. Considering research on social skill development and advancements in portable technology, a mobile application that can support anytime-anywhere support to children with this deficit is timely.

EYEdentify is a game that uses a basic learner model to provide a flexible intervention in the form of an engaging game. It has an open learner model that can show the child's progress to parents, caregivers, teachers, and specialists. The first version of this application incorporates four emotions (i.e., happy, sad, frustrated, and confused) into a matching game that progresses through different levels (Fig. 1). There are three types of images that are used in this game to help scaffold the child's learning: cartoon robot faces, real faces that are superimposed on robot faces, and photographs of actual faces. The cartoon robot faces are designed to emphasize the eyes and mouth. The superimposed faces are designed to activate the child's knowledge of focusing on the eyes and mouth to correctly identify the displayed emotions while maintaining the scaffold of the robot head. The photograph of an individual making a particular expression is used to activate the knowledge from the previously superimposed images to correctly identify the emotions. Difficulty increases with respect to the type of emotion that is incorporated into game play and the types of images that are used. Positive feedback is provided to the child throughout the game to encourage continuous play. The game also has a calming event that is triggered by the accelerometer when the mobile device is shaken aggressively. The calming event increases the volume of the music that is being played and prompts the child to count to ten. The child is then asked whether or not s/he wants to continue playing the game.

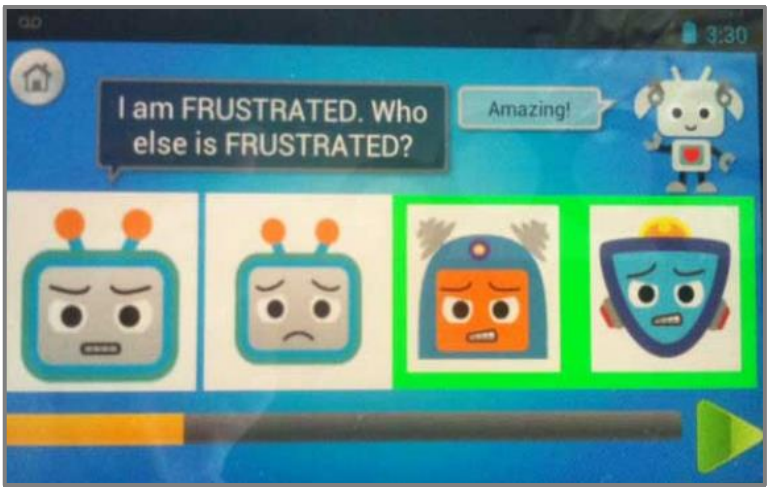


Fig. 1. The gameplay screen with the correct responses identified (surrounded in green).

The mobile application provides the ability to customize game play by incorporating personalized feedback and images. Users can customize feedback by typing a comment and recording an audio message before adding this feedback to the schedule. Image customization uses the front camera of the device to capture individuals parroting the facial expression represented on the robot prompt. As children progress through the levels, they are rewarded with parts to assemble their own robot.

The current version focuses on developing emotion recognition skills for four of the fifteen basic emotions identified by Golan et al. [25]. The addition of the remaining eleven emotions could be used to extend game play. Currently, the mobile application is functional; however, more emotions are being incorporated and iOS versions are being developed before releasing EYEdentify on Google Play and the App Store.

5.2 Expanding EYEdentify to Include a Simulated Learning Environment

The expansion of EYEdentify to include a simulated learning environment draws on Csikszentmihalyi's definition of flow and research on gaming. Flow is described as the experience of being fully engaged in an activity where an individual is “so involved…that nothing else seems to matter” [27]. This is derived from activities where a person’s skills are matched to the challenges encountered [27]. For learners, this means that they will be in a mental state that keeps them motivated to stay involved in a particular activity. Research in gaming and game design incorporates these psychological underpinnings whereby elements of a game seek to cultivate and support the player’s active engagement and enhanced motivation [28]. In educational games, these elements are employed to scaffold learning just-in-time and provide instructors with the ability to adapt the system to the specific needs of the learner [29].

EYEdentify currently provides a matching game with rewards that are self-contained within the mobile application. Preliminary trials indicate that it keeps learners involved in the activity of identifying emotions for long periods of time. These trials parallel the findings of research that used a video intervention program known as "The Transporters" to develop the social skills of children with ASD [30].

EYEdentify's game play can be expanded into simulated learning environments to move players beyond the acquisition of emotion recognition skills toward the development of social skills. In creating game-based simulations for learners to use, the capacity to scaffold their learning within game play and support the development of transferable skills to the real-world increases.

There are several ways to expand game play into a simulated learning environment. All possibilities would require the mastery of basic emotion recognition and could involve levels of progressive difficulty that incorporates these emotions into depictions of social situations. The front camera of the mobile device could be used to scaffold the recognition of emotions by way of augmented reality, as could the recent introduction of Google glass. Avatars that represent individuals from the learner's day-to-day life could be used by learners to practice particular social situations. Additionally, game play could incorporate depictions of situations that model different social interactions. This could then be incorporated with a Sims-like environment where learners would have to identify the emotion of the character that they are interacting with and demonstrate the appropriate behaviour or emotional response. Specific to keeping learners engaged, the addition of an emotion recognition system that can detect the learner's emotion from the front camera and keep track of their emotion when playing the game to determine that learner's level of engagement would be useful. Through the development of these possibilities, EYEdentify has the potential to enhance learners' emotion recognition and social skill development in a way that enables the learner to transfer these skills to their day-to-day encounters.

6 Conclusion

The use of simulated learners and learning environments within special education contexts holds great potential for improving the quality and applicability of ITS use by members of this population. Simulated learners can be used to test learning materials, learning methods, and ITS to ensure their appropriateness for the members of this population, who have highly variable needs. The use of simulated learners and learning environments can be further exploited for teacher training. In addition to this use, simulated learning environments can be used to help learners who have been classified as having special needs to transfer their knowledge and skills to their everyday lives. The potential for members of this population to use simulated learning environments was illustrated through an example of an educational game, EYEdentify, that is used to help children with autism spectrum disorder improve their ability to recognize emotions. The described potential expansions of this game show how different approaches to simulated learning environments and the use of augmented reality can be used to help learners transition between the simulated world and the one they encounter every day.

References

1. Bruno, A., Gonzalez, C., Moreno, L., Noda, M., Aguilar, R., Munoz, V.: Teaching mathematics in children with Down's syndrome. In: Artificial Intelligence in Education (AIED). Sydney, Australia (2003).
2. Government of Ontario: Individual Education Plans Standards for Development, Program Planning, and Implementation. Ontario Ministry of Education (2000).
3. Nilsson, A., Pareto, L.: The complexity of integrating technology enhanced learning in special math education - a case study. In: 5th European Conference on Technology Enhanced Learning on Sustaining TEL: from Innovation to Learning and Practice. pp. 638–643. Springer-Verlag, Berlin, Heidelberg (2010).
4. Wainer, A.L., Ingersoll, B.R.: The use of innovative computer technology for teaching social communication to individuals with autism spectrum disorder. Research in Autism Spectrum Disorders. 5, 96–107 (2011).
5. Assessment, equity, and opportunity to learn. Cambridge University Press, Cambridge ; New York (2008).
6. Jackson, L.A., Witt, E.A., Games, A.I., Fitzgerald, H.E., von Eye, A., Zhao, Y.: Information technology use and creativity: Findings from the Children and Technology Project. Computers in Human Behavior. 28, 370–376 (2012).
7. Carnahan, C., Basham, J., Musti-Rao, S.: A Low-Technology Strategy for Increasing Engagement of Students with Autism and Significant Learning Needs. Exceptionality. 17, 76–87 (2009).
8. Tentori, M., Hayes, G.: Designing for Interaction Immediacy to Enhance Social Skills of Children with Autism. Ubiquitous Computing (Ubicomp). pp. 51–60. ACM, Copenhagen, Denmark (2010).
9. Moffatt, K., Findlater, L., Allen, M.: Generalizability in Research with Cognitively Impaired Individuals. In: Workshop on Designing for People with Cognitive Impairments, ACM Conference on Human Factors in Computing Systems (CHI). ACM, Montreal, Canada (2006).
10. Woolf, B.P., Arroyo, I., Muldner, K., Burleson, W., Cooper, D.G., Dolan, R., Christopherson, R.M.: The Effect of Motivational Learning Companions on Low Achieving Students and Students with Disabilities. In: Aleven, V., Kay, J., and Mostow, J. (eds.) Intelligent Tutoring Systems (ITS). pp. 327–337. Springer Berlin Heidelberg, Berlin, Heidelberg (2010).
11. Cardon, T.A., Wilcox, M.J., Campbell, P.H.: Caregiver Perspectives About Assistive Technology Use With Their Young Children With Autism Spectrum Disorders. Infants & Young Children. 24, 153–173 (2011).
12. Code of Fair Testing Practices in Education. Washington, D.C.: Joint Committee on Testing Practices, American Psychological Association (1988).
13. VanLehn, K., Ohlsson, S., Nason, R.: Applications of Simulated Students: An Exploration. International Journal of Artificial Intelligence in Education (IJAIED). 5, 135–175 (1996).
14. Mertz, J.S.: Using A Simulated Student for Instructional Design. International Journal of Artificial Intelligence in Education (IJAIED). 8, 116–141 (1997).

15. Dawe, M.: Design Methods to Engage Individuals with Cognitive Disabilities and their Families. In: the Science of Design Workshop, ACM Conference on Human Factors in Computing Systems (CHI) (2007).
16. Paramythis, A., Weibelzahl, S., Masthoff, J.: Layered Evaluation of Interactive Adaptive Systems: Framework and Formative Methods. User Modeling and User-Adapted Interaction (UMUAI). 20, 383–453 (2010).
17. Rogers, M., Hwang, H., Toplak, M., Weiss, M., Tannock, R.: Inattention, working memory, and academic achievement in adolescents referred for attention deficit/hyperactivity disorder (ADHD). Child Neuropsychology. 17, 444–458 (2011).
18. Geary, D.C.: Mathematics and Learning Disabilities. J Learn Disabil. 37, 4–15 (2004).
19. Ogan, A., Finkelstein, S., Mayfield, E., D'Adamo, C., Matsuda, N., Cassell, J.: "Oh dear stacy!": social interaction, elaboration, and learning with teachable agents. In: ACM Conference on Human Factors in Computing Systems (CHI). pp. 39–48. ACM, New York, NY, USA (2012).
20. Moreno, R., Mayer, R., Lester, J.: Life-Like Pedagogical Agents in Constructivist Multimedia Environments: Cognitive Consequences of their Interaction. In: World Conference on Educational Multimedia, Hypermedia and Telecommunications (EDMEDIA). pp. 776–781 (2000).
21. Henderson-Summet, V., Clawson, J.: Usability at the Edges: Bringing the Lab into the Real World and the Real World into the Lab. In: Workshop on Usability in the Wild, International Conference on Human-Computer Interaction (INTERACT) (2007).
22. Van Dam, A., Becker, S., Simpson, R.M.: Next-generation educational software: why we need it & a research agenda for getting it. EDUCAUSE Review. 40, 26–43 (2007).
23. Stromer, R., Kimball, J.W., Kinney, E.M., Taylor, B.A.: Activity schedules, computer technology, and teaching children with autism spectrum disorders. Focus on Autism and Other Developmental Disabilities. 21, 14–24 (2006).
24. DiGennaro Reed, F.D., Hyman, S.R., Hirst, J.M.: Applications of technology to teach social skills to children with autism. Research in Autism Spectrum Disorders. 5, 1003–1010 (2011).
25. Golan, O., Ashwin, E., Granader, Y., McClintock, S., Day, K., Leggett, V., Baron-Cohen, S.: Enhancing Emotion Recognition in Children with Autism Spectrum Conditions: An Intervention Using Animated Vehicles with Real Emotional Faces. Journal of Autism and Developmental Disorders. 40, 269–279 (2009).
26. Erickson, K., Schulkin, J.: Facial expressions of emotion: A cognitive neuroscience perspective. Brain and Cognition. 52, 52–60 (2003).
27. Csikszentmihalyi, M.: Flow: The Psychology of Optimal Experience. Harper Perennial Modern Classics (2008).
28. Tom Chatfield: 7 ways games reward the brain | Video on TED.com. (2010).
29. Fernández López, Á., Rodríguez Fórtiz, M.J., Noguera García, M.: Designing and Supporting Cooperative and Ubiquitous Learning Systems for People with Special Needs. In: Confederated International Workshops and Posters on the Move to Meaningful Internet Systems: ADI, CAMS, EI2N, ISDE, IWSSA, MONET, OnToContent, ODIS, ORM, OTM Academy, SWWS, SEMELS, Beyond SAWSDL, and COMBEK. pp. 423–432. Springer-Verlag, Berlin, Heidelberg (2009).
30. The Transporters. Changing Media Development Ltd (2006).

Simulated Students, Mastery Learning, and Improved Learning Curves for Real-World Cognitive Tutors

Stephen E. Fancsali, Tristan Nixon, Annalies Vuong, and Steven Ritter

Carnegie Learning, Inc.
Frick Building, Suite 918
437 Grant Street, Pittsburgh, PA 15219

{sfancsali, tnixon, avuong, sritter}
@carnegielearning.com

Abstract. We briefly describe three approaches to simulating students to develop and improve intelligent tutoring systems. We review recent work with simulated student data based on simple probabilistic models that provides important insight into practical decisions made in the deployment of Cognitive Tutor software, focusing specifically on aspects of mastery learning in Bayesian Knowledge Tracing and learning curve analysis to improve cognitive (skill) models. We provide a new simulation approach that builds on earlier efforts to better visualize aggregate learning curves.

Keywords: Knowledge tracing, learning curves, student modeling, Cognitive Tutor, simulation, simulated students, mastery learning

1 Introduction

There are at least three general approaches to simulating students for the purposes of improving cognitive (skill) models and other features of intelligent tutoring systems (ITSs). One approach, generally connoted in discussions of "simulated" students or learners, employs aspects of cognitive theory to simulate students' learning and progression through ITS problems (e.g., via machine learning or computational agents like SimStudent [2]). Another class of simulations makes use of relatively simple probabilistic models to generate response data (i.e., Bayesian Knowledge Tracing [BKT] [1]) intended to represent a (simulated) student's evolving performance over many practice attempts. Third, there are data-driven approaches that do not easily fit into either of the first two categories.

In this work, we explicate and provide examples of each approach and briefly describe Carnegie Learning's Cognitive Tutors (CTs) [3]. We then focus on the second approach and review recent work on simulations of student learning with simple probabilistic models. These simulation studies provide novel insights into a variety of features of CTs and their practical deployment.

CTs implement mastery learning; mathematics content is adaptively presented to students based upon whether the tutor has judged that a student has mastered particular skills. Mastery is assessed according to whether the tutor judges that the probability that a student has mastered a particular skill exceeds a set threshold. We review a simulation study that provides for best and worst-case analyses (when "ground truth" characteristics of simulated learner populations are known) of tutor skill mastery judgment and efficient student practice (i.e., adaptively providing students with opportunities to practice only those skills they have not mastered). This study not only provides justification for the traditionally used 95% probability threshold, but it also illuminates how the threshold for skill mastery can function as a "tunable" parameter, demonstrating the practical import of such simulation studies.

Finally, learning curves provide a visual representation of student performance on opportunities to practice purported skills in an ITS. These representations can be used to analyze whether a domain has been appropriately atomized into skills. If opportunities correspond to practice for a single skill, we expect to see a gradual increase in the proportion of correct responses as students get more practice opportunities. If, for example, the proportion of students responding correctly to an opportunity drastically decreases after three practice opportunities, it seems unlikely that the opportunities genuinely correspond to one particular skill. Turning to the third, data-driven approach to simulating students, we provide a new method to visualize aggregate learning curves to better drive improvements in cognitive (skill) models used in CTs, This approach extends recent work that explores several problems for utilizing learning curves aggregated over many students to determine whether practice opportunities correspond to a single skill.

2 Cognitive Tutors

CTs are ITSs for mathematics curricula used by hundreds of thousands of K-12 and undergraduate students every year. Based on cognitive models that decompose problem solving into constituent knowledge components (KCs) or skills, CT implements BKT to track student skill knowledge. When the system's estimate of a student's knowledge of any particular skill exceeds a set threshold, the student is judged to have mastered that skill. Based on the CT's judgment of skill mastery, problems that emphasize different skills are adaptively presented so that the student may focus on those skills most in need of practice.

3 Three Approaches to Simulating Learners

There are at least three general simulation methods used to model student or learner performance. One simulation strategy, based on cognitive theories such as ACT-R [4], explicitly models cognitive problem-solving processes to produce rich agent-based simulated students. The SimStudent project ([2], [5]), for example, has been developed as a part of a suite of authoring tools to develop curricula for CTs, called Cognitive Tutor Authoring Tools (CTAT) [6]. SimStudent learns production rules from problem-solving demonstrations (e.g., an author providing simple demonstrations of problem solutions or via ITS log data). These human-interpretable production rules correspond to KCs that comprise cognitive models vital to CTs. SimStudent aims to simplify development of new CT material by automating the discovery of KC models in new domains via a bottom-up search for skills that potentially explain the demonstrations.

Second, there are numerous probabilistic methods that model task performance as a function of practice, according to various task and learner-specific parameters. One may instantiate numerous such models, with varying parameters, and sample from the resulting probability distributions to obtain simulated performance data for an entire hypothetical learner population.

One common example is a Hidden Markov Model (HMM) with two latent and two observable states, that can serve as a generative BKT model, using parameters specified according to expert knowledge or inferred by a data-driven estimation procedure. Two hidden nodes in the HMM represent "known" and "unknown" student knowledge states. In practice, of course, student knowledge is latent. Simulated students are assigned to a knowledge state according to BKT's parameter for the probability of initial knowledge, $P(L_0)$, and those in the "unknown" state transition to the "known" state according to the BKT parameter for the probability of learning or transfer, P(T). Simulated, observed responses are then sampled according to BKT parameters that represent the probability of student guessing, P(G) (i.e., responding correctly when in the unknown state) and slipping, P(S) (i.e., responding incorrectly when in the known state), depending upon the state of student knowledge at each practice opportunity.

Contrary to her real-world epistemological position, simulations generally allow an investigator to access the student's knowledge state at each simulated practice opportunity. This allows for comparisons between the "ground truth" of skill mastery and any estimate derived from resulting simulated behavior. Clearly, richer cognitive agents, such as SimStudent, provide a more complete picture of the student's cognitive state at any point.

Simpler probabilistic models represent student knowledge of a skill with a single state variable, so they correspondingly scale better to larger scale simulations of whole populations. While a probabilistic model only requires a reasonable distribution over initial parameters, richer cognitive models may require training on a great deal of detailed, behavioral or demonstration data. Nevertheless, cognitive model-based simulations allow us to investigate issues like timing (i.e., response latency), sensitivity to input characteristics, and error patterns in learner responses.

There are many cases in which a relatively simple probabilistic model may be of utility, despite its impoverished nature. A simplistic representation of student knowledge provides an ideal situation to test the performance and characteristics of inference methods using data from a known generating process and parameters. One might, for example, compare the point at which simulated students acquire knowledge of a skill to the point at which the CT judges the student to have mastered the skill. The approach thus allows for students of "best" and "worst" case scenarios with respect to the relationship between how the CT models students and the actual make up of (simulated) student populations. We can better understand the dynamics of the student sub-populations we inevitably face in practice by simulating data from diverse sub-populations, the make up of which we can specify or randomize in various ways. Furthermore, we can simulate student performance (sometimes augmenting available empirical data) both with and without mastery learning (i.e., students being removed from a population because they have mastered a skill) on learning curves constructed from aggregate data.

Previous work [7] explored a third, data-driven simulation method that "replays" empirical student performance data through CT in order to estimate the impact of a change in BKT parameters in a more substantive way. For each KC that occurred in a given problem, we sampled the next observed response on that KC from the sequence actually observed from a real student. These responses would then drive updates to CT's cognitive model, knowledge tracing, and the problem-selection mechanism. If more data were required than were observed for a given student, further observations were sampled from a BKT model initialized to the state inferred from the student's actions thus far. By repeating this process for all students in the observed data set, we could obtain estimates of the number of problems students would be expected to complete if a change to the cognitive model were implemented.

This method has the advantage of preserving characteristics of real student data rather than resorting to a theoretical model of student performance. However, it does make several assumptions about the reproducibility of that behavior under the hypothesized changes. Specifically, it assumes that the observed sequence of correct/incorrect responses would not change even given a different selection of problems, potentially emphasizing different KCs. This assumption may be justified if we believe we have complete coverage of all KCs relevant to the task in question in the cognitive model and that all KCs are truly independent of each other.

While simulation methods based on rich cognitive theory and data-driven re-play of empirical data provide many opportunities for future research, we focus in this paper on simple, probabilistic simulations in the context of the BKT framework.

4 Substantive Measures of Efficient Student Practice

Before we discuss how the BKT mastery threshold probability functions as a "tunable" parameter in an ITS like the CT, we provide "substantive" quantification of goodness of fit of cognitive/skill models for CTs beyond mere RMSE of prediction (i.e., beyond the extent to which models can predict whether students will respond correctly to particular practice opportunities) [8-11]. New error or goodness of fit measures are countenanced in terms of efficient student practice, based on the number of practice opportunities (i.e., "over-practice" or "under-practice") we might expect a student to experience in a CT. Over-practice refers to the continued presentation of new practice opportunities, despite the student's mastery or knowledge of the relevant KC.[1] Student "under-practice" instances are those in which a student has yet to

[1] One exception is an experimental study [11] that reports increased efficiency by deploying parameters estimated using a data mining method called Learning Factors Analysis (LFA).

achieve knowledge of a KC, and yet the mastery learning system has judged the student as having mastered it, ending the presentation of further learning opportunities. From estimates of expected under- and over-practice, one can calculate other meaningful measures of students gains and losses, such as time saved or wasted.

Some of this work [8, 9] uses empirical data to estimate the extent of under-practice and over-practice we might expect students to experience. Specifically, the expected numbers of practice opportunities it takes a student to reach mastery when parameters are individualized per student are compared to the expected practice when a single (population) set of parameters is used to assess all students. One individualization scheme used to study under and over-practice estimates all four BKT parameters, per student, from response data over all relevant skills (i.e., each student receives one individualized quadruple of BKT parameters for all KCs) [8]. Another approach [9] only individualizes P(T) for each student based on both per-student and per-skill components estimated from observed data [12]. Both individualization schemes provide for substantive gains (compared to using a set of population parameters to assess all students' progress to mastery) in the efficiency of practice (i.e., fewer expected under and over-practice opportunities) as well as better prediction performance judged, in the standard way, by a metric like RMSE.

5 Idealized Performance of Mastery Learning Assessment

Now we address how BKT performs with respect to efficiency of practice in idealized cases in which the composition of student (sub-) populations is known. Simulation studies can shed light on how BKT performs when mastery learning parameters used by the CT run-time system exactly match those of the generating model (i.e., the best case), and in worst cases in which student parameters either maximally differ from mastery learning parameters or vary at random for each student.

Recent work addresses these issues by adopting a probabilistic simulation regime [10]. Since we can track the point at which a simulated student acquires knowledge of a skill, we are able to compare this to the opportunity at which the mastery learning system first judges it to be acquired. Simulations were run for fourteen skills, a subset of those found by [13] to be representative of a substantial portion of skills in deployed CT curricula, across thousands of virtual students.

Even in idealized, best case scenarios (i.e., when parameters used to assess skill mastery perfectly match simulated student data-generating parameters), for most skills and a large number of students, we expect there to be one to four "lagged" practice opportunities between the point at which simulated students transition to mastery and the point at which the BKT run-time system judges mastery. That is, in general, even when a student population is modeled "perfectly," and given the traditional setting of the probability threshold for mastery at 95%, most students should be expected to see at least a few opportunities beyond the point of skill acquisition. That some "over-practice" may be inevitable provides a relevant context within which to consid-

Efficiency is operationalized as decreased time required to work through material in the Geometry CT without decreasing overall learning.

er empirically driven results of [8, 9]. Although a certain amount of lag may be inherent in the nature of BKT, we seek to establish a range for the "acceptable" lag, and to better appraise efficiency of practice [10].

6 Mastery Learning Threshold as a "Tunable" Parameter

In addition to lagged opportunities and over-practice, situations in which students under-practice skills are important to consider. Given the possibly inevitable lag between skill acquisition and mastery judgment, simulations [10] have also been used to explore how the mastery probability threshold might be "tuned" to influence the trade-off of over-practice and under-practice experienced by students in mastery learning systems like CTs.

Pre-mature mastery judgments can lead, for example, to students being moved along by the CT to problems that emphasize new KCs without having mastered prerequisite KCs. Other things held equal, simulations in [10] provide that pre-mature mastery judgment is more likely to occur in worst-case scenarios, when mastery-learning parameters do not match parameters for sub-populations of simulated students.

Simulations in [10] also establish that the mastery-learning threshold can function as a tuning parameter, partially governing the trade-off between the expected proportion of students pre-maturely judged to have reached skill mastery and the number of over-practice opportunities they are likely to experience. As the threshold probability is increased, the proportion of students assessed as having pre-maturely mastered skills decreases while the proportion of those that are exposed to practice opportunities after skill acquisition increases (along with the number of lagged and over-practice opportunities, i.e., those beyond a calculated acceptable lag they experience).

The results of [10] show that the traditionally used 95% threshold seems to provide for a "conservative" tutor that is more likely to present opportunities after skill acquisition rather than under-practice. Depending upon course design and practice regimes, the mastery-learning threshold might be manipulated to important, practical effect. For example, pre-mature mastery judgments might be acceptable in larger numbers when there is a mixed-practice regime that will allow students to practice KCs later in the curriculum.

7 Using Simulations to Illuminate Learning in Learning Curves

Learning curves provide a visual representation of student performance over opportunities to practice skills. For each (purported) skill, we construct a learning curve by plotting opportunities (i.e., 1st, opportunity, 2nd opportunity, and so on) on the x-axis and the proportion of students that provide correct responses at each opportunity on the y-axis. Aggregated over real-world student practice opportunity data, such curves provide means by which to visually[2] inspect whether opportunities genuinely correspond to practice of one particular skill. If opportunities correspond to one particular skill, we expect a gradual increase in the proportion of students that respond correctly with increasing practice. Generally, for well-modeled skills (and a variety of other cognitive tasks), it is thought that such a plot should correspond roughly to a power law function (i.e., the power law of practice [14]), though this point is not without controversy [15]. Recent research [16-17] demonstrates how some aggregate learning curves can distort the picture of student learning. Aggregate learning curves may, for example, appear to show no learning, when, in fact all students are learning at different rates. Others may provide for a small rise in probability of correct response initially but then "drop," as if students were forgetting, even when individual students are consistently mastering their skills.

The learning curve of Fig. 1 illustrates aspects of both problems, with a relatively flat portion, followed by a drop, after a small increase in probability correct from its initial value. The red line, representing the size of the student population at each opportunity, illustrates that BKT is determining that students are mastering the skill relatively quickly.

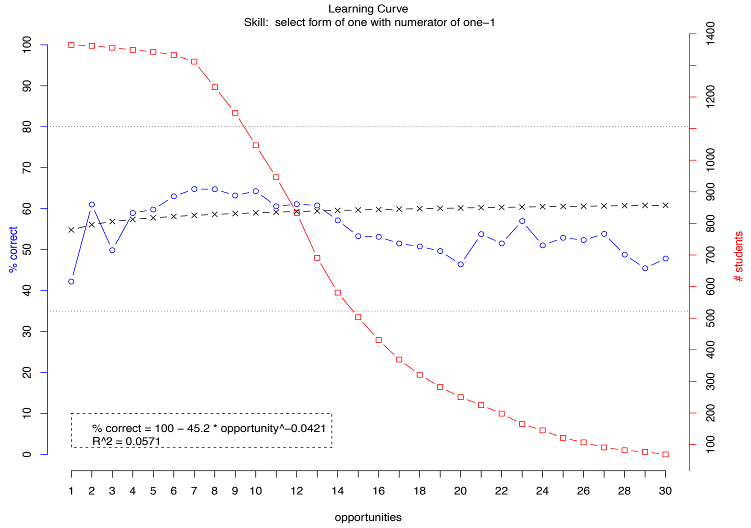


Fig. 1. Empirical Learning Curve for Skill "Select form of one with numerator of one"; the blue line represents empirical data plotted as percentage of correct responses, and the black line represents a fitted power function. The red line provides the size of the student population.

Two ways to re-visualize problematic, aggregated learning curves have been suggested [16]. One is to provide multiple learning curves (on the same plot) for individual

[2] Developers at Carnegie Learning also deploy several data-driven heuristics (that correspond to various visual features of learning curves) to analyze our large portfolio of KCs (i.e., several thousand KCs over several mathematics CT curricula) and observed student data to draw attention to those KCs that may require revision in our deployed cognitive models.

"segments" of students based upon how many opportunities students, in observed data, take to reach the mastery learning threshold for a skill. Such segmented learning curves are provided with the same x-axis and y-axis as standard learning curves (i.e., practice opportunity count on the x-axis and, e.g., percentage of student correct response on the y-axis).

The second approach suggested by [16] has the analyst plot "mastery-aligned" learning curves. In such learning curves, students are also segmented according to the number of opportunities required to reach mastery, but the end-point of the x-axis corresponds to the opportunity at which students' reach mastery (*m*) and moving left along the x-axis corresponds to the opportunity before mastery (*m-1*), the second to last opportunity before mastery (*m-2*), and so on.

Further work [17] provides a mathematical explanation, along with proof-of-concept simulation studies based on HMMs, for the dynamics of aggregate learning curves to explain how both mastery learning itself and differing student subpopulations, when aggregated, can contribute to learning curves that do not show learning (or manifest other peculiar, possible deceptive, phenomena like "negative" learning).

We illustrate an alternative to [16] by providing a method that relies on probabilistic simulation to construct aggregate learning curves that better represent learning in empirical student data. Specifically, we "pad" empirical data for student skill opportunities with simulated data to mask the effects of attrition due to mastery learning and possibly "reveal" student learning. Student opportunity data are generated with the same parameters used to track student progress and the probability of student knowledge estimated at the point at which the student crossed the mastery threshold. Such simulations provide us data after a student no longer receives practice opportunities for a particular skill because they have been judged as having achieved mastery.

For the aggregate learning curve of Fig. 1, the "padded" learning curve is Fig. 2. The fitted power-law slope parameter decreases from -0.042 to -0.363 (indicating more learning), and the goodness-of-fit of the power law function (R^2) increases from 0.0571 to 0.875. We apply the method to 166 skills identified[3] by [16] as possibly problematic in the Cognitive Tutor Algebra I (CTAI) curriculum. We find an improvement (i.e., power-fit parameter decreases from above -0.1 to below -0.1, a criterion deployed by [16]) for 98 skills (59%). While this method provides an improved visualization and understanding of fewer skills than the disaggregation procedures suggested by [16], this seems to provide evidence of the great extent to which mastery learning attrition obfuscates evidence for student learning.

Importantly, our simulation method does not eliminate the early dip in the learning curve at opportunity 3 when little attrition has yet to take place, but only masks the effects of attrition due to mastery learning. Such an approach focuses largely on a better representation or visualization of the "tail" of aggregate learning curves. This

[3] These skills were chosen because the over-whelming majority of students are judged to eventually master them (i.e., CT "thinks" the students are learning); they are not pre-mastered (i.e., $P(L_0) < 0.95$); they do not show learning in their aggregate learning curve (i.e., power-law fit parameter > -0.1); aggregate learning curves for these skills do not have multiple maxima; and we have data for at least 250 students for these skills [16].

allows us to focus on other features of the learning curve that may indicate ill-modeled KCs in a cognitive model, software bugs, and other possible problems.

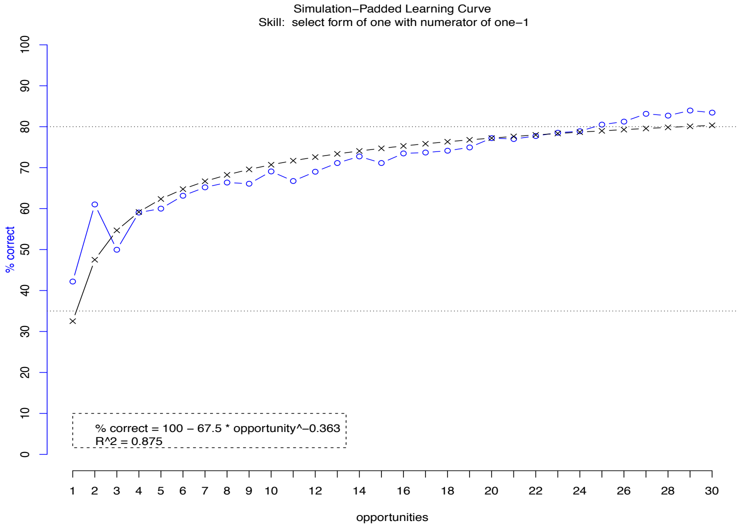


Fig. 2. Simulation-Padded Learning Curve for Skill "Select form of one with numerator of one"

8 Summary

We briefly reviewed several methods for simulating learners. We focused on ways in which simple probabilistic models, in contrast to methods that rely on rich cognitive theory, can be used to generate student performance data to help drive practical decision-making about CT deployment, focusing first on the mastery threshold probability of BKT as a tunable parameter to determine aspects of efficient practice. Then we introduced a new method for visualizing aggregate learning curves that relies on both empirical and simulated data that helps to mask the bias introduced by mastery learning attrition. Future work will further explore these methods, new simulation regimes, and their practical import.

References

1. Corbett, A.T., Anderson, J.R.: Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. User-Modeling and User-Adapted Interaction 4, 253–278 (1995)
2. Matsuda, N., Cohen, W.W., Sewall, J., Koedinger, K.R.: Applying Machine Learning to Cognitive Modeling for Cognitive Tutors. Human-Computer Interaction Institute, Carnegie Mellon University. Paper 248 (CMU-ML-06-105) (2006)

3. Ritter, S., Anderson, J.R., Koedinger, K.R., Corbett, A.T.: Cognitive Tutors: Applied Research in Mathematics Education. Psychonomic Bulletin & Review 14, 249–255 (2007)
4. Anderson, J.R.: Rules of the Mind. Erlbuam, Hillsdale, NJ (1993)
5. Matsuda, N., Cohen, W.W., Sewall, J., Lacerda, G., Koedinger, K.R.: Evaluating a Simulated Student Using Real Students Data for Training and Testing. In: Proceedings of the International Conference on User Modeling (LNAI 4511), pp. 107–116 (2007)
6. Aleven, V., McLaren, B.M., Sewall, J., Koedinger, K.R.: The Cognitive Tutor Authoring Tool (CTAT): Preliminary Evaluation of Efficiency Gains. In: Proceedings of the 8th International Conference on Intelligent Tutoring Systems, pp. 61-70 (2006)
7. Dickison, D., Ritter, S., Nixon, T., Harris, T., Towle, B., Murray, R.C., Hausmann, R.G.M.: Predicting the Effects of Skill Model Changes on Student Progress. In: Proceedings of the 10th International Conference on Intelligent Tutoring Systems (Part II), pp. 300-302 (2010)
8. Lee, J.I., Brunskill, E.: The Impact of Individualizing Student Models on Necessary Practice Opportunities. In: Proceedings of the 5th International Conference on Educational Data Mining, pp. 118–125 (2012)
9. Yudelson, M.V., Koedinger, K.R.: Estimating the Benefits of Student Model Improvements on a Substantive Scale. In: Proceedings of the 6th International Conference on Educational Data Mining (2013)
10. Fancsali, S., Nixon, T., Ritter, S.: Optimal and Worst-Case Performance of Mastery Learning Assessment with Bayesian Knowledge Tracing. In: Proceedings of the 6th International Conference on Educational Data Mining (2013)
11. Cen, H., Koedinger, K., Junker, B.: Is Over-Practice Necessary? – Improving Learning Efficiency with the Cognitive Tutor through Educational Data Mining. In: Proceedings of the 13th International Conference on Artificial Intelligence in Education, pp. 511–518 (2007)
12. Yudelson, M.V., Koedinger, K.R., Gordon, G.J.: Individualized Bayesian Knowledge Tracing Models. In: Proceedings of the 16th International Conference on Artificial Intelligence in Education (2013)
13. Ritter, S., Harris, T.K., Nixon, T., Dickison, D., Murray, R.C., Towle, B.: Reducing the Knowledge Tracing Space. In: Proceedings of the 2nd International Conference on Educational Data Mining, pp. 151-160 (2009)
14. Newell, A., Rosenbloom, P.S.: Mechanisms of Skill Acquisition and the Law of Practice. In: Anderson, J.R. (ed.) Cognitive Skills and Their Acquisition, pp. 1-55. Erlbaum, Hillsdale, NJ (1981)
15. Heathcote, A., Brown, S.: The Power Law Repealed: The Case for an Exponential Law of Practice. Psychonomic Bulletin & Review 7, 185-207 (2000)
16. Murray, R.C., Ritter, S., Nixon, T., Schwiebert, R., Hausmann, R.G.M., Towle, B., Fancsali, S., Vuong, A.: Revealing the Learning in Learning Curves. In: Proceedings of the 16th International Conference on Artificial Intelligence in Education, pp. 473-482 (2013)
17. Nixon, T., Fancsali, S., Ritter, S.: The Complex Dynamics of Aggregate Learning Curves. In: Proceedings of the 6th International Conference on Educational Data Mining (2013)

Exploring through Simulation the Effects of Peer Impact on Learning

Stephanie Frost[1] and Gord McCalla[1]

ARIES Lab, Dept. of Computer Science, U. of Saskatchewan, Saskatoon, Canada
stephanie.frost@usask.ca, mccalla@cs.usask.ca

Abstract. Simulation modelling helps designers to keep track of many possible behaviours in a complex environment. Having a technique to simulate the effect of peer impact on learning allows designers to test the social effects of their educational software. We implement an agent-based simulation model based on the ecological approach (EA) architecture [9]. The model considers learner attributes, learning object attributes and two styles of peer impact to explore the effects when learners are either positively or negatively impacted by high achieving peers. In this study, we observe different patterns of behaviour based on the style of peer impact and by limiting simulated learners' access to information (the EA metadata). Gaining understanding of these patterns will inform our future work on recommending sequences of learning objects (LOs).

Keywords: simulated learning environments, simulated learners, ecological approach, instructional planning

1 Introduction

Before taking an action in a learning environment, it is important for an intelligent tutoring system (ITS) to have some way of estimating the likelihood that the action will be successful, i.e. that it will benefit the learner(s) involved. To compute such an estimate, there are many dimensions to consider such as: the nature of the content being learned, the pedagogical style of the environment, learning goals, individual learner characteristics, and social factors such as how a learner's own performance can be influenced by knowledge of peer performance. Such complexity is often managed through the use of models.

Simulation modelling can be used by instructional developers for testing their systems; this was identified by VanLehn, Ohlsson and Nason [11] in a survey of possible uses of simulated students. One example is SimStudent by Matsuda et al. [8] which can be used by designers to explore through simulation the effects of various decisions on cognitive tutor design. Whether a model is used "internally" (by an ITS to compute the next action) or "externally" (to evaluate a system design), a challenge remains: How does the model estimate the amount of learning that occurs when a learner interacts with a Learning Object (LO)? In particular, we wanted to explore the impact on learning when learner performance is influenced by the performance of peers. Some learners may become encouraged when observing high peer achievement and perform even better than they would have otherwise. Other learners might become discouraged in the same situation and perform even worse. Having a technique to simulate the effects of peer performance would allow instructional developers to test social effects of their designs. In this paper, we use simulation to explore the behaviours exhibited by two different reactions to peer impact.

We describe our approach in Section 2, followed by the simulation study in Section 3. It is possible to simulate many different kinds of educational software in the ecological approach (EA) architecture [5], and then test the simulation under various conditions to get insight into issues the designer is interested in. Because our model is implemented in the EA architecture, our approach for modelling peer impact can be used across many different styles of learning systems. The data to feed our simulation is synthetic, but could, itself, be modelled on data extracted from actual learner behaviour [5]. We follow with a description of ongoing research that uses simulation for testing and developing a method for recommending sequences of LOs, and conclude with a discussion of our findings.

2 Model Structure

In another paper [5], we have argued that it is not necessary to model every detail of the learning process, but that systems can be tested in a simulation that captures only the most relevant characteristics for a given purpose. Therefore, we take an approach that lets an instructional developer choose different dimensions – such as attributes of the learning objects, aspects of the pedagogical environment, attributes of the learner – and assign weights to each dimension according to the priorities of the developer. This section describes the structure of the simulation model so as to provide background for the experiment around peer impact, described in Section 3.

The EA architecture [9] provides a way to record metadata about learner interactions with LOs. As learners interact with LOs, any information that is known about the learner at the time of the interaction can be saved as metadata and associated with the LO. The EA assumes that each learner is represented by a learner model that contains static attributes (*characteristics*) as well as other data gathered as they interact with the LOs (*episodic*).

We developed an agent-based simulation model with very simple abstractions of learners and LOs. Each learner agent has an attribute, *aptitude-of-learner*, a number between (0,1), which we use to model the range of aptitudes (low to high) different learners have for a given subject matter. In our model, this attribute is assigned at the start of the simulation and does not change, but in future work we plan to create more sophisticated simulations where this attribute is not static. The simulated LOs have an attribute to represent *difficulty level*, which is also a number between (0,1) where higher values represent more difficult material. The simulated LOs are arranged into a random directed acyclic graph to represent prerequisite relationships between the LOs.

The model execution revolves around an atomic action: the learner's interaction with a LO. This action might occur hundreds or thousands of times during a simulation run, thus creating a multitude of EA metadata from which measurements can be taken. In related work [5], we introduce the term *evaluation function* to describe the function that computes the degree of success as result of an interaction between a learner and a LO. We will use the term *P[learned]* to describe the value that is generated by the evaluation function, i.e. the "probability that the learner learned the LO", or the "system's belief that the learner knows the LO". The P[learned] value is included as part of the EA metadata that is associated with LOs after learners interact with them.

Our evaluation function is a weighted sum, where each term deals with a dimension of learning to be considered. Each dimension of learning is calculated with a mini function. For example, suppose Learner$_{\mathrm{A}}$ were a novice with *aptitude-of-learner*=0.1. Next, suppose LO$_{\mathrm{X}}$ were a fairly easy LO, which implies a high probability of success. We use a mini function, *difficulty-of-LO*, to translate the LO difficulty attribute into a high probability value, giving *difficulty-of-LO*=0.8. Suppose we also wish to take into account that the likelihood of the learner learning the LO is higher if the learner has already viewed prerequisite LOs. Prerequisite information is given in the LO attributes. Our simulation model has a function for *hasPrerequisites* which searches through the EA metadata to discover whether the learner has indeed viewed the prerequisites and returns 1.0 if the answer is yes and 0.0 otherwise. If we want these dimensions to have approximately equal weights, then we can define the evaluation function below and obtain P[learned] as follows:

$$\begin{gathered}(\mathrm{w})(\text{aptitude-of-learner}) + (\mathrm{w})(\text{difficulty-of-LO}) + (\mathrm{w})(\text{hasPrerequisites}) \\ = (0.33)(0.1) + (0.33)(0.8) + (0.34)(1.0) = 0.637\end{gathered}$$

If, on the other hand, we wish to give the aptitude a higher weight, such as 60%, then the new value could be (0.6)(0.1) + (0.2)(0.8) + (0.2)(1.0), or 0.42. As expected, giving greater weight to this learner's low aptitude decreases the P[learned] somewhat. More dimensions can be incorporated so long as the weights sum to 1.0. The evaluation function, implemented as a weighted sum, will provide an estimated likelihood the LO has been learned between (0,1), making it easy to compare averages of such P[learned] values between simulation runs. However, we caution against comparing two simulation runs with different evaluation functions (i.e. different weights or dimensions) because that would be like comparing two numbers with different units of measure.

The independent variables in our experiment are the *aptitude-of-learner* values, the *difficulty level* values, the directed acyclic graph giving prerequisite relationships between LOs, as well as a dimension called *peer-impact*, which is explained in the next section.

3 Experiment

Our experiment is intended to explore through simulation the effects of peer impact on learning. We motivate the experiment by visiting literature around how peers can impact each other's scores.

Students are impacted by their peers even in their ordinary lives. A study was performed by Hanushek et al. [6] to clarify the impacts of peer group characteristics on achievement in the context of family and school factors, race and socio-economic status. Results suggested that students benefitted from higher achieving schoolmates. In contrast, the American Academy of Pediatrics warned that Facebook pages can make some children feel badly because they see themselves as being inferior to their peers [10]. This effect is due to the nature of Facebook, where most users will censor their posts and only share the most positive information about themselves, skewing the view of reality. Along the same lines, Daniel et al. [3] found in a study that learners will usually only participate in online learning activities if they have trust in their peers or some degree of self confidence.

Others have used simulations to study peer effects. Mao et al. [7] used a simulation model to study the impact of social factors in a course where students shared learning materials with each other. The output of Mao et al.'s model was a comparison of the amount of sharing connected to status levels: gold, silver, bronze, common. Populations fluctuated as users began at the common status and gradually transitioned between levels. The paper concluded that simulation models can be useful for developing and improving incentive mechanisms in virtual communities. In a different study, Zhang et al. [12] studied the fluctuation of a population of learners through various activities: registration, activation, action and adaptation. The authors found that learners who participated the most were also the ones most sensitive to changes in the community and had the most fluctuations.

This research, and other research, shows that a learner's score can be impacted by peer performance. We decided to explore this issue by creating a notion of "peer impact", where learners respond differently from one another according to how well other learners are doing in mastering the LOs. This takes the form of a new dimension in our evaluation function called *peer-impact*. Like the other dimensions we discussed in Section 2 (*aptitude-of-learner*, *difficulty-of-LO*, *hasPrerequisites*), this is a function that produces a value between (0,1) to represent a positive or negative impact on P[learned]. In our experiment, we use the following Equation 1 to compute P[learned] each time a learner visits a LO.

$$.25(\text{apt-of-learner}) + .25(\text{diff-of-LO}) + .25(\text{hasPrereq}) + .25(\text{peer-impact}) \quad (1)$$

We created two styles of peer impact called *reinforcing* and *balancing* which refer to a comparison between an individual learner's average P[learned] on the LOs they have viewed so far, compared to the average P[learned] of all learner agents, which we call "class average". The information to compute these

P[learned] averages is obtained from the EA metadata. Each learner is given one of these styles at the start of the simulation and it remains fixed. Future work could explore more sophisticated learner agents where this attribute is not static.

The reinforcing style means that the learner's score is "attracted" to the class average P[learned]. That is, when the class average is higher than their own, the peer impact function for a reinforcing learner produces a value close to 1; thus the learner will perform even better than they would have otherwise. This is a positive feedback loop, because as the learner performs better so does the class average thus further encouraging the learner to do better. If the class average is lower than their own, then the *peer-impact* function gives a value close to zero; thus the learner will do even worse than they would have otherwise.

Balancing is the opposite. In this case, a learner's score is "repelled" from the class average P[learned]. That is, when the class average is higher than the individual's average P[learned], then their score will be pulled down lower than it would have been otherwise. This is a negative feedback loop because when the class average is high, the learner's average goes in the other direction. When the class average is low, then the learner's score will be boosted higher than it would have otherwise. In Figure 1, we show the *peer-impact* function (the values 0.2 and 0.8 were chosen as thresholds to allow clear effects of the two types of learner to emerge).

```
if currentLearner BALANCING
   if class average is HIGHER than mine
      set peerImpact == randomNumBetween(0.0,0.2)
   if class average is LOWER than mine
      set peerImpact == randomNumBetween(0.8,1.0)
if currentLearner REINFORCING
   if class average is HIGHER than mine
      set peerImpact == randomNumBetween(0.8,1.0)
   if class average is LOWER than mine
      set peerImpact == randomNumBetween(0.0,0.2)
```

Fig. 1. Function to generate *peer-impact* for a given learner at a given time in the simulation

The dependent variable in our experiment is the P[learned] values generated by the simulation; we gain insight into whether the peer impact has a positive or negative effect by observing the relative P[learned] values. We varied this experiment under six conditions. We varied the proportions of balancing and reinforcing styles: mostly balancing, mostly reinforcing, and fifty-fifty. For instance, if the model is set to mostly balancing, when new learners are initialized, they have a high chance of being assigned the balancing personality and a low chance of being assigned the reinforcing personality. These three proportions were each run under two difficulty levels: one with mostly easy LOs and high aptitude learners, and the other with mostly difficult LOs and low aptitude learners. These six conditions were hand picked to be representative samples on a curve of possible population mixes that should provide some insight about the effect of these two kinds of personality on the learning environment. We ran each of the six conditions 5 times because our model is stochastic; it produces slightly different results each time even under the same starting conditions.

A typical result is shown in Figure 2 (fifty-fifty, high difficulty with low aptitude learners). Each line represents the average P[learned] of different portions of the simulated learner population: the lightest thin line for all learners, black thin line for the learners who were assigned the reinforcing personality, and the dark grey thin line for the learners who were assigned the balancing personality. Normally, our simulation model would be used to evaluate a particular instructional planning technique, but because this experiment is intended to illuminate peer impact, the order in which LOs are consumed isn't important. Therefore, the simulated learners, of which there are 80, visited random LOs, of which there are 100.

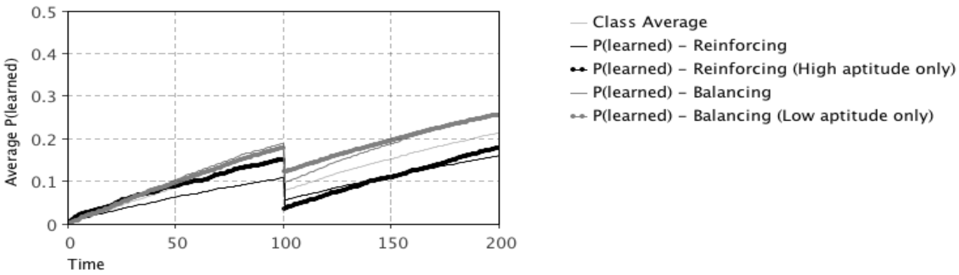


Fig. 2. Typical result

At the start of the simulations, the class average starts at zero. The balancing simulated learners had higher scores in this state because this is the behaviour defined in the evaluation function – that balancing learners do well when the class average is lower than their individual average. The learning gradually increases for both groups as the simulated learners visit more and more LOs. Although the results seem low overall – P[learned] only reaching short of 0.3 – this is due to the number of LOs (100) created in the simulation and the time it would take for learners to visit them all. We ran the simulation again with only 30 LOs and observed the same patterns, but with a steeper slope; the average P[learned] reached around 0.5. This raises interesting questions about whether the amount of time required to learn a set of LOs should actually be represented with a linear function. In reality, learners would get tired or lose interest or change their learning goals. Future work could compare instructional plans with learners having different levels of stamina.

The thick lines in Figures 2 and 3 represent subsets of the balancing and reinforcing personalities whose behaviour we wish to discuss in this experiment. Simulated learners do not have access to the actual class average, but compute the average based on what other simulated learners have allowed them to perceive about their performance. Based on Daniel et al.'s [3] results that confident learners are more likely to share their success, simulated learners with high P[learned] values shared their EA metadata, while those with lower P[learned] values did not. This creates a *suppression effect*, where each simulated learner has access to different information in the computation of how others are doing, depending on which other learners have suppressed information at the time they are computing the average.

The thick grey line shows only the balancing learners with low aptitudes while the thick black line shows only the reinforcing learners with high aptitudes. At the start of the simulation, the thick black line is below the thick grey line: it is perhaps surprising that a group of simulated learners with high aptitudes would have overall lower scores than a group of simulated learners with low aptitudes. We highlight this because it shows that different parts of the evaluation function – *peer-impact*, *aptitude-of-learner* etc. – can dominate at different times. In this case, high aptitude can be dominated by peer impact for reinforcing personalities when the class average is low.

In Figure 3, we observe another interesting phenomenon by injecting 80 more simulated learners halfway though the experiment, a somewhat contrived situation, although one that might happen in the real world if, say, two classes merged partway through a course, or if two study groups in an online course were mashed together, or due to the openness of many online courses (e.g. MOOCs) when new learners can join any time. Under most of the experimental conditions we tried, such as the typical result in Figure 2, although the influx of new learners caused the class average to drop (as expected, because each new learner starts with an average P[learned] of zero), there was no apparent change in the relative ranking of the groups of learners being measured. That is, if the balancing learners had the highest average before the influx, this continued afterward. However, in about a third of the runs with low difficulty LOs and high aptitude learners, the influx of learners caused a *phase shift*: now the thick black line jumps above the thick grey line (see Figure 3). This makes sense: the balancing learners who tend to do more poorly when the class average drops, do just that. The influx also creates a situation where there are now learners with high averages intermingled with learners with zero averages; this creates a different environment than the starting condition where everyone started at zero. Different environmental conditions cause the model to exhibit different behaviour. With the suppression effect deactivated, all learners have access to the same information. In this condition, we observed that the thick grey line overlapped with the thick black line and there was no apparent phase shift (i.e. no lines crossing over).

Even though the observed patterns are merely a result of the evaluation function implementation – that is, the model is simply doing what it was programmed to do – it helps system designers to keep track of the different possible

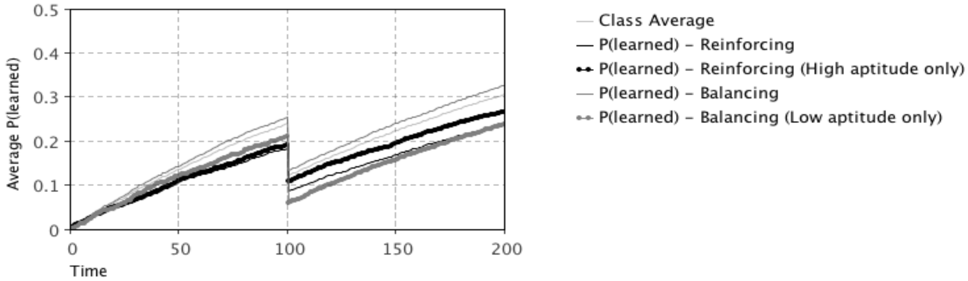


Fig. 3. Condition showing phase shift

behaviours as they try to design systems to support learning in all of these conditions: low or high aptitude learners, easy or difficult material, peer effects, prerequisites and many other possible dimensions, with each behaving differently in different situations. Without simulation, it is unlikely we would have made our observations about the phase shift as well as the observation about the high aptitude reinforcing learners having lower scores than low aptitude balancing learners. These observations reveal the specific circumstances that instructional developers should address in order to maximize the expected learning. For example, the system could be programmed to intervene when it detects that the current class average will push a learner's expected outcome in an undesirable direction. When the class average is higher than an individual's average, the scores of other learners should be displayed more prominently for the balancing learners but not for the reinforcing learners.

Through this experiment, we have also shown that simulations can be used to test unexpected situations. Future experiments could test for influxes of new LOs instead of new learners. Other variations could look at adding or removing LOs to impact the difficulty level of the course or the level of expertise of peer learners. When we injected a herd of simulated learners, we observed some surprising results. But, by examining the underlying dynamic behaviour as the simulation proceeded, we could actually explain why these results happened, thus gaining more intuition about learning that would help to better inform an experiment that might be carried out with real learners.

4 Other Research Directions

In ongoing work, we are also developing a technique for recommending sequences of LOs. Instructional planners have been built that explore different kinds of sequencing such as sequencing things of the same type, like "lessons" or even sequencing several types of activities, like presentations and assessments [1]. Our method involves using the EA metadata to identify "trails" of LOs. We are investigating the use of user-based and item-based approaches to generate recommendations of these trails using Apache Mahout [1]. Using information captured in the EA metadata, we create metrics for giving sequences a score to reflect the quality of the sequence, for example does P[learned] increase or decrease over the sequence. We are also exploring changes to the evaluation function to favour sequences that suggest coherence, such as trails that give learners a view of the big picture before going into the details. Sequences with high scores are then used as a basis for recommending sequences to other learners. Our study will examine whether learners receiving sequence recommendations see any improvement over learners receiving one LO recommendation at a time.

Other work in simulating recommender systems for learning systems has been done by Drachsler et al. [4]; but the main difference is that this work did not involve sequences, peer impact or the EA architecture. Champaign [2] uses the ecological approach architecture to use the experiences of past learners to suggest sequences of LOs for future learners while also studying the impact of peer ratings, which are not the same as our peer impact because our peer impact is linked to the evaluation function.

Even with the simplistic models of learners and LOs we have presented so far, the peer impact experiment demonstrates the combinatorics of the various features is already becoming too complex to rely on human intuition; this is one of the main reasons for simulation modelling.

5 Conclusion

We created simulated learners whose overall learning was influenced by one of two styles of peer impact. Our study demonstrated that different patterns emerge when when simulated learners change their own behaviour based on the behaviour of the group and when these learners have limited access to information due to others' ability to suppress their EA metadata. In some conditions, a phase shift occurred from the initial situation where the class average is zero to a new situation with some learners having relatively high averages. The simulated learners prior to the influx had higher averages because they had the opportunity to visit LOs before the arrival of the new simulated learners. One style of peer impact is not universally better or worse than another, but each has advantages in different circumstances. It is important for instructional developers to understand such patterns. In future work, the use of simulations with the EA architecture will shed more light on peer impact and will allow us to also factor in the effects of different kinds of sequence recommendations.

The EA metadata make it easy to look deeply into the underlying dynamics and identify the conditions that create such behaviours. The EA metadata also allow us to change the inputs of the simulation and take measurements, as we did to compare the P[learned] averages between learners with different styles of peer impact. By using the EA architecture for the simulation studies, the later construction of a real learning system is made easier if the real system also uses

[1] http://mahout.apache.org/

the EA architecture. That is, if the real system also stores information about a learner's interaction with a LO as metadata associated with the LO, then estimating the likelihood of success for a real learner follows the same methods used by developers to estimate the success of simulated learners.

Acknowledgements
We would like to thank Dr. Julita Vassileva for discussions with the first author about using simulations for instructional planning and for insights on applying social concepts in the simulation model; Graham Erickson for his ideas about the evaluation function; and Dr. Nathaniel Osgood for discussions about cause loop diagrams and agent-based modelling in AnyLogic. We also wish to acknowledge the Natural Sciences and Engineering Research Council of Canada for funding some aspects of this research through a Discovery Grant to the last author.

References

[1] Brusilovsky, P. and Vassileva, J.: Course sequencing techniques for large-scale web-based education. International Journal of Continuing Engineering Education and Lifelong Learning, 13, 75-94 (2003).

[2] Champaign, J.: Peer-based intelligent tutoring systems: a corpus-oriented approach. Ph.D. Thesis, University of Waterloo, Waterloo, Canada (2012)

[3] Daniel, B., McCalla, G., Schwier, R.: Social Network Analysis techniques: implications for information and knowledge sharing in virtual learning communities. Int. J. of Interactive Media in Education, 2(1), 20-34 (2008)

[4] Drachsler, H., Hummel, H., and Koper, R.: Using simulations to evaluate the effects of recommender systems for learners in informal learning networks. In Learning, 3 CEUR Workshop Proc., 404-423 (2008).

[5] Erickson, G., Frost, S., Bateman, S., and McCalla, G.: Using the ecological approach to create simulations of learning environments. To appear in Lane, H.C., Yacef, K., Graesser, A., Mostow, J. (eds.), Proc. 16th Int. Conf. on AIED, Memphis (2013)

[6] Hanushek, E., Kain, J., Markman, J., Rivkin, S.: Does peer ability affect student achievement? Journal of Applied Economics, 18, 527-544 (2003)

[7] Mao, Y., Vassileva, J., and Grassmann, W.: A system dynamics approach to study virtual communities. In Proc. 40th Annual Hawaii Int. Conf. on System Sciences, HICSS '07, pp.178a-, Washington, DC, USA, IEEE Computer Society (2007)

[8] Matsuda, N., Cohen, W.W., Sewall, J., Lacerda, G. and Koedinger, K.R.: Predicting students performance with SimStudent that learns cognitive skills from observation. In Luckin, R., Koedinger, K.R., and Greer, J. (eds.), Proc. 12th Int. Conf. on AIED, Marina del Rey, 467-476 (2007)

[9] McCalla, G: The ecological approach to the design of e-learning environments: purpose-based capture and use of information about learners. Journal of Interactive Media in Education.

[10] Tanner, L: Docs warn about teens and 'Facebook depression'. Associated Press. http://www.msnbc.msn.com/id/42298789/ns/health-mental_health/t/docs-warn-about-teens-facebook-depression/ Accessed April 13, 2013.

[11] VanLehn, K., Ohlsson, S., and Nason, R.: Applications of simulated students: an exploration. Int. J. Artificial Intelligence in Education, 5, 135-175 (1996)

[12] Zhang, Y., and Tanniru, M.: An agent-based approach to study virtual learning communities. Hawaii International Conference on System Sciences, 1(11c) (2005)

Using HCI Task Modeling Techniques to Measure How Deeply Students Model

Sylvie Girard, Lishan Zhang, Yoalli Hidalgo-Pontet, Kurt VanLehn,
Winslow Burleson, Maria Elena Chavez-Echeagary, Javier Gonzalez-Sanchez

Arizona State University, Computing, Informatics, and Decision Systems Engineering, Tempe, AZ, 85281, U.S.A.

{sylvie.girard, lzhang90, yhidalgo, kurt.vanlehn, winslow.burleson, helenchavez, javiergs}@asu.edu

Abstract: User modeling in AIED has been extended in the past decades to include affective and motivational aspects of learner's interaction in intelligent tutoring systems. An issue in such systems is researchers' ability to understand and detect students' cognitive and meta-cognitive processes while they learn. In order to study those factors, various detectors have been created that classify episodes in log data as gaming, high/low effort on task, robust learning, etc. When simulating students' learning processes in an ITS, a question remains as to how to create those detectors, and how reliable their simulation of the user's learning processes can be. In this article, we present our method for creating a detector of shallow modeling practices within a meta-tutor instructional system. The detector was defined using HCI (human-computer interaction) task modeling as well as a coding scheme defined by human coders from past users' screen recordings of software use. The detector produced classifications of student behavior that were highly similar to classifications produced by human coders with a kappa of .925.

Keywords: intelligent tutoring system, shallow learning, robust learning, human-computer interaction, task modeling

1 Introduction

Advances in student modeling in the past two decades enabled the detection of various cognitive [3, 4, 8, 11, 13, 16, 17], meta-cognitive [1,6], and affective [2, 9] processes during learning based on classification of episodes in log data. Steps have been taken toward detecting when learning occurs [4] and to predict how much of the acquired knowledge students can apply to other situations [5, 6]. However, an obstacle in such research is how to gain an understanding of the user's cognitive or meta-cognitive processes while learning. While some of the indicators used in the literature are common to any intelligent tutoring system, others are closely linked to the activities and pedagogical goals of a specific application. The adaptation of such indicators to the design of a new system often necessitates a detailed analysis of the new domain and how the tutoring system guides learners to acquire its skills and knowledge. In particular, an issue within this process is the ability to reach common ground between learner scientists that perform an analysis of learners (meta-)cognitive actions at a high level - via video or log analysis of student's past actions for example – and the definition of the indicators by software engineers, related to how the system was implemented, that can be used to simulate such processes in agreement with the constraints and functionalities of software. We view the specificity of detectors as unavoidable, so the best solution is to develop good methods for analyzing the new tutoring system and designing the detectors. This short article describes our method and its application to out project, AMT. In the AMT project, a choice was made to use HCI (human computer interaction) task modeling - a method for formally representing human activity, and by extension, the behavior of an interactive system -, as well as video coding schemes from human coders, to develop the detectors. The detectors aim to evaluate student's use of shallow and deep modeling practices with and without being guided by a meta-tutor, on the domain of dynamic systems modeling.

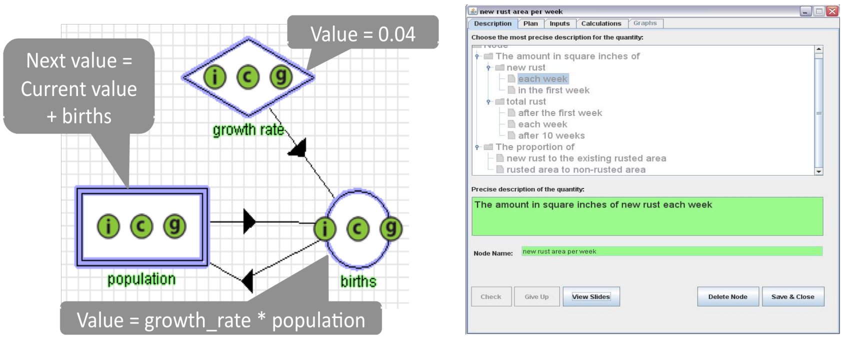
In Section 2, the AMT learning environment, for which the detectors were created, is introduced. In a third section, the task model of the user's activity in AMT is described. Next, the process of defining a coding scheme for the detector with human coders is presented, followed by the definition of the different classifications that define the value, the implementation and empirical evaluation of the detector. The final section summarizes the uses of task modeling within this work, and how it could be applied in future to other applications.

2 AMT software: a meta-tutor to teach deep modeling of dynamic systems.

AMT software teaches students how to create and test a model of a dynamic system. In our modeling language, a model is a directed graph with one type of link, as illustrated in Figure 1. Each node represents both a variable and the computation that determines the variable's value. There are three types of nodes.

- A *fixed value* node represents a constant value that is directly specified in the problem. A fixed value node has a diamond shape and never contains incoming links.
- An *accumulator* node accumulates the values of its inputs. That is, its current value is the sum of its previous value plus or minus its inputs. An accumulator node has a rectangular shape and always has at least one incoming link.
- A *function* node's value is an algebraic function of its inputs. A function node has a circular shape and at least one incoming link.

The students' learning objective is to draw a model representing a situation that is described in the form of a relatively short text. In the example of Figure 1, the description of the problem was " *Rust destroys steel and can spread quickly. Suppose you take a large sheet of steel, such as one that might be used as the roof of the box-car on a train, and you put it outside in the weather. Suppose it starts with a spot of rust that is 10 square inches in area. However, each week the rust spot gets bigger, as it grows by 30%. Therefore at the end of the first week, the rust spot is 13 square inches in area.*" and the objective of the problem was to "Graph the size of the rust spot over 10 weeks."



Choose the most precise description for the quantity:

- The amount in square inches of
  - new rust
    - each week
    - in the first week
  - total rust
    - after the first week
    - each week
    - after 10 weeks
- The proportion of
  - new rust to the existing rusted area
  - rusted area to non-rusted area

Precise description of the quantity:

The amount in square inches of new rust each week

Node Name: new rust area per week

Fig. 1. The left image is the example of model, with gray callouts added to explain the contents of nodes. The right image is the example of a node editor.

The student constructs the model node by node, by filling in all information within each node in the form of four interactive tabs (description, plan, inputs, and calculations). During construction, students can use the *Check* button to evaluate the correctness of the current tab, or the *Solve it for me* button to ask the system to fill out the tab automatically.

The instruction is divided into three phases: (1) an introduction phase where students learn basic concepts of dynamic system model construction and how to use the interface; (2) a training phase where students are guided by a tutor and a meta-tutor to create several models; and (3) a transfer phase where all scaffolding is removed from soft-ware and students are free to model as they wish. The tutor gives feedback and corrections on domain mistakes.

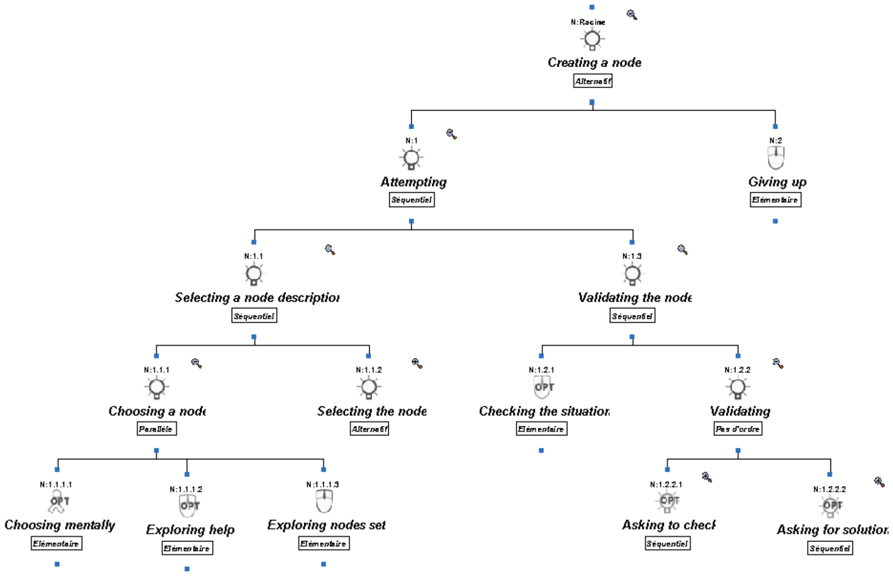
The meta-tutor requires students to follow a goal-reduction problem solving strategy, the Target Node Strategy [18]. The basic idea is to focus on one node at a time (the target node) and completely define it before working on any other node. This process decomposes the whole problem of modeling a system into a series of atomic modeling problems, one per node. Like *Pyrenees* [2], it teaches students that if they just master this one difficult but small skill, then the rest of the problem solving will be straight-forward. In addition, the meta-tutor complains if students appear to be guessing too much or giving up too early, just as the *Help Tutor* did [3].

While students learn, their motivation, attention to details, and modeling depth can fluctuate. To assess students, the project needed detectors that detect shallow and deep modeling practices both with and without the meta-tutor. The measure should be usable in the transfer phase of the experiment as a dependent variable, because deep modeling is the skill/knowledge that AMT teaches. The depth measure should also apply to student's behavior during the training phase so that we can check whether the instructional manipulations done during that phase have their intended effects (i.e., the measure serves as a manipulation check). The detector should further operate in real time (i.e., it doesn't require to know future actions or states in order to interpret the current action) so that it can be eventually be used by the system itself to condition its behavior.

3 Task Modeling: analysis of user's actions on software

A task model is a formal representation of the user's activity. It is represented by a hierarchical task tree to express all sub-activity that enables the user to perform the planned activity. The tasks need to be achieved in a specific order, defined in the task tree by the ordering operators. In AMT, every modeling activity follows the same procedure involving the same help features, task flow, and meta-tutor interventions. With a single task model of a prototypical modeling task, it is therefore possible to account for all of the user's activity in software. Due to the complexity of the final model, only one sub-activity will be described in this paper, illustrated in Figure 2. Only part of the model is deployed in the figure, and some subtasks will not be detailed here. In this part of the model the sub-activity the learner wishes to perform is to create a new node for the dynamic system s/he is currently modeling. We will first describe the task tree, and then insert the iterations and conditions that enable a formal verification of the flow of the task within the task model.

Figure 2: Sub-task "Creating a Node" in the AMT activity task model using K-MADe



Short description of the sub-task to model:

In order for a node to be created, the description tab of the node editor needs to be completed by selecting a node description, which corresponds to a valid quantity in the system to model. Each node is unique and cannot be created more than once. The user can engage in the task only if at least one node still needs to be created for the model to be complete.

Task tree and order of the tasks:

At the top level of the task tree "Creating a node", the learner can either attempt to create the node (task 1) or give up on the creation (task 2). The second task is represented in software by the user closing the node editor window, and can be done at any time during the task. The task "Creating a node" is over when a good description has been found and validated. The system can then try to initialize the selection and create the node.

In the first level of the task "Attempting", the learner first needs to select a node description (task 1.1), i.e.: what quantity the node will represent. S/he is then allowed to finish the creation of the node by validating the selection (task 1.2).

In order to select a node description, the user first needs to choose a node description (task 1.1.1) among the set of node descriptions offered by the system. This process involves the user choosing mentally one description (task 1.1.1.1), exploring the help features offered by software (task 1.1.1.2) and exploring the set of node descriptions displayed (task 1.1.1.3). S/he can then select the node (task 1.1.2). This subtask is not described in Figure 1 for a lack of space.

In order to validate the selection, the learner can choose to go back to the description of the problem to verify the correctness of his solution according to the problem to be simulated (task 1.2.1), and then has to validate the selection (task 1.2.1.2). When the user checks the validity of the selection, it can either be performed by checking the solution against the set of nodes still remaining to be modeled (task 1.2.1.2.1) or asking software to produce the solution (task 1.2.1.2.2). The user is allowed to ask for the solution only when a description has been checked at least once.

Now that the different actions of the learner are defined, the iterations and conditions will help represent the flow of the activity on the subtask "Selecting a node description" (task 1.1).

Iterative and Optional tasks

- Task 1.1 is iterative: it is possible to make several selections before trying to finish the description by validating.
- Task 1.1.1.2 is optional: The learner is not forced to explore the help features to choose a description, this is merely a choice on the learner's part.
- The main task, "creating a node", is iterative until the node is created or the activity is abandoned. The later is represented in the task model by an interruptible task: the learner can stop his/her creation of node activity any time by choosing to close the node editor window.

Conditions on tasks:

- Main task 1 has a pre-condition attached to it: the software only allows the user to engage in a creation of a new node if there is at least one node related to the modeling of the dynamic system that still remains to be created.

A first task model was created to represent learner's activity on software without the presence of the meta-tutor. This corresponds to the first version of software, which was evaluated against the interface including the meta-tutor in [18]. This second software interface includes a text-based agent that intervenes as the students engage in modeling to help them achieve deeper modeling behaviors, by applying constraints to the user's actions and giving meta-cognitive feedback. The meta-tutor was therefore added to the task model under the type "system" and the model was completed to include the constraints and interventions of the meta-tutor.

The final task model produced represented all possible actions of the learner on software in order to model a dynamic system. Next, a study of these actions, which led to the definition of the depth detectors, is detailed.

4 Detecting when students are modeling using shallow practices

The task model developed with K-MADe was used to define the episode structure. The first step in creating a coding scheme is to define a unit of measurement for the user's modeling actions. The task model clearly highlighted the different sub-activities the learner could engage in, referred to as goals. All goals are interruptible tasks in favor to accessing the help features[1] or abandoning the completion of the current goal for a new one. After a brainstorming session where researchers studied how students' actions fell in line with those goals, the following unit of depth, called "segment", was defined. This established the unit of coding to be used in the next phase.

Screen videos representing the learners' use of the AMT software with and without the meta-tutor were recorded during an experimental study described in [6]. These videos were studied to determine how much shallow vs. deep modeling occurred and the contexts, which tended to produce each type. A coding system was then created for video recordings of the learners' behavior. Three iterations of design for this coding scheme were performed, ending with a coding scheme that reached a multi-rater pairwise kappa of .902. The final coding scheme mapped learners' behavior to six classifications, which were implemented as the following depth detectors[AIED short paper]

- GOOD_METHOD: The students followed a deep method in their modeling. They used the help tools appropriately, including the one for planning each part of the model.
- VERIFY_INFO: Before checking their step for correctness, students looked back at the problem description, the information provided by the instruction slides, or the meta-tutor agent.

[1] It is to be noted that two help systems are available to users: (1) referring back to the instructions always available for viewing, and (2) looking at the problem situation where all details of the dynamic system to model are described.

- SINGLE_ANSWER: The student's initial response for this step was correct, and the student did not change it.
- SEVERAL_ANSWERS: The student made more than one attempt at completing the step. This includes guessing and gaming the system:
    - The user guessed the answer, either by clicking on the correct answer by mistake or luck, or by entering a loop of click and guessing to find the answer.
    - The user "games the system" by using the immediate feedback given to guess the answer: series of checks on wrong answers that help deduce the right answer.
- UNDO_GOOD_WORK: This action suggests a modeling misconception on the students' part. One example is when students try to run the model when not all of the nodes are fully defined.
- GIVEUP: The student gave up on finding how to do a step and clicked on the "give up" button.

Another detector was defined as a linear function of the six episode detectors. It was intended to measure the overall depth of the students' modeling, therefore providing an outcome measure in the transfer phase in future experimental studies. It considered two measures (GOOD_ANSWER, VERIFY_INFO) to indicate deep modeling, one measure (SINGLE_ANSWER) to be neutral, and three measures (SEVERAL_ANSWERS, UNDO_GOOD_WORK, and GIVE_UP) to indicate shallow modeling.

Once the coding scheme reached a sufficient level of agreement between coders, the task model was used to adapt the coding to students' actions on the software. The episodes that were coded for depth by human analysts in the sample video were analyzed by creating scenarios from the task model within K-MADe. The validation of six detectors' implementation involved three human coders, who watched a sample of 50 episodes, paying attention to the depth of modeling exhibited by the student's actions, and chose the classification that best represented the depth of the learner modeling at the time of the detected value. A multi-rater and pairwise kappa was then performed, reaching a level of inter-reliance of .925.

5 The different uses of the Task Model

The task modeling language K-MAD and its task model creation and simulation environment, K-MADe [7] were chosen for the following reasons: the environment enables the creation and replay of scenarios of student's actions, a set of functionalities not described here enable a formal verification of the model. Additionally the associated simulation environment ProtoTask [14] allows non-specialists in task modeling to visualize the flow of the task model, via scenarios in a clear and simple manner.

The use of K-MAD helped in the creation of the detectors and are a first step in offering an alternative technique to simulated learners, by tackling the following problems:

- *Breaching the gap between learner scientists' understanding of how the learning process works and programmers' definition of the application flow, functionalities, and indicators.*
- *Enabling a formal validation of software flow, understandable by all.*
- *Using simulated learners scenarios to define the detectors.*

A researcher in educational technology - expert in teaching modeling and part of the AMT project - and an HCI practitioner, realized the task model. The former was an expert on how AMT software was designed in terms of pedagogical content and task flow. His expertise focused in particular on the actions the students were allowed/incited/forbidden to do within software at each moment of the modeling task. The HCI practitioner was not familiar with intelligent tutoring systems or meta-tutors. She was involved in the creation of the task model in a consulting capacity, in regards to her expertise in task modeling of interactive systems.

The task model could be defined at the level of the user's planning of actions and system flow, with iterations and conditions alone. However, the objects in K-MADe enable us to represent the constraints of the learner's actions concretely and to apply a formal verification of task flow. It was therefore possible to represent the set of descriptions as either valid or invalid, to detect when a node has been checked and the result of that check, and to add constraints on the checking procedure such as to avoid node duplication. This enabled a formal verification of software flow prior to validate its fidelity to learner scientists' ideas about possible actions on software and the underlying processes involved.

Once the model was constructed, the use of ProtoTask to visualize software flow and follow learners' possible sets of actions allowed by software enabled the ability to simulate learners by creating scenarios of use that could be played and replayed at will, focusing on the cognitive and meta-cognitive levels of learner's experience on software. In the process of creating our detectors, a video analysis of learner's past actions was performed. The model could be used to check the possible actions of users with what the designer of the system wanted to offer as functionalities and software flow. During this analysis, the task model could be used once again to define scenarios that simulated learner's pertinent behaviors using ProtoTask. Once those scenarios were formed, the task analyst came back to the original K-MAD modeling language and studied the similarities and contrasts between scenarios to define the rules that govern the detection of shallow and deep modeling practices within AMT. Once the task model identified points of detection of such practices, it became easy for programmers to go back to software and implement the rules.

6 Conclusion and Future Work

In this paper, a method to create a detector of deep modeling within a meta-tutor using HCI task modeling and video coding schemes was described. The main outcome of this process was the creation of detectors inferring the depth of students' modeling practices while they learn on a meta-tutoring system, reaching a multi-rater and pairwise kappa score of .925. We believe the use of the task model to define shallow and deep modeling practices by helping to create the detectors to be of value for any simulated learning environments, in particular for indicators that a common to all learning tasks present in a tutoring system.

In interdisciplinary teams, the design of indicators can lead to communication issues due to misunderstandings and a lack of common ground between analysis made at a high level of learners' cognitive and meta-cognitive processes, and the representation of those behaviors within software. In particular, video-coding processes can become costly when the coders' understanding of the details of how the system works differs from how the system actually works. Our experience using K-MADe and ProtoTask highlighted an ease in this project in gaining a better view of the tutoring system and the detection of deep modeling within the interface. In particular, the use of ProtoTask by the non-specialists in task modeling helped clarify issues of task flow and the definition of the set of user's actions at each moment of interaction.

A limitation of the method is the applicability to different types of tutoring systems. In AMT, a single task model was able to represent the entirety of a users' learning activity. In tutoring systems that teach a set of skills through different pedagogical approaches for diverse types of learning tasks, the creation of such task models might prove more costly and may not be completely adapted to the creation of detectors that need to be adapted to each task specifically.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 0910221. We would like to thank Sybille Caffiau for consulting in the project and sharing her expertise in task modeling of interactive systems.

References

1. Aleven, V., McLaren, B.M., Roll, I., Koedinger, K.R.(2006): Toward meta-cognitive tutoring: A model of help seeking with a Cognitive Tutor. International Journal of Artificial Intelligence and Education 16, 101–128
2. Arroyo, I., and Woolf, B.P., 2005. Inferring learning and attitudes from a Bayesian Network of log file data. In Proceedings of the 2005 conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology, Chee-Kit Looi, Gord McCalla, Bert Bredeweg, and Joost Breuker (Eds.). IOS Press, Amsterdam, The Netherlands, The Netherlands, 33-40.
3. Baker, R. S. J. d., Corbett, A. T., Koedinger, K. R., Evenson, S., Roll, I., Wagner, A. Z., … Beck, J. E. (2006). Adapting to when students game an intelligent tutoring system, Proceedings of the 8th international conference on Intelligent Tutoring Systems, Jhongli, Taiwan Berlin, Heidelberg.
4. Baker, R.S.J.d., Goldstein, A.B., Heffernan, N.T.: Detecting the Moment of Learning. In: Aleven, V., Kay, J., Mostow, J. (eds.) ITS 2010. LNCS, vol. 6094, pp. 25–34. Springer, Heidelberg (2010)
5. Baker, R. S. J. D., Gowda, S. M., & Corbett, A. T. (2011). Towards predicting future transfer of learning, Proceedings of the 15th international conference on Artificial intelli-

gence in education. Proceedings from AIED'11, Auckland, New Zealand Berlin, Heidelberg.

6. Baker, R. S. J. D., Gowda, S. M., Corbett, A. T., & Ocumpaugh, J. (2012). Towards automatically detecting whether student learning is shallow., Proceedings of the 11th international conference on Intelligent Tutoring Systems, Chania, Crete, Greece Berlin, Heidelberg.
7. Caffiau, S., Scapin, D., Girard, P., Baron, M., & Jambon, F. (2010). Increasing the expressive power of task analysis: Systematic comparison and empirical assessment of tool-supported task models. Interacting with Computers, 22(6), 569–593. doi:10.1016/j.intcom.2010.06.003
8. Corbett, A.T., MacLaren, B., Kauffman, L., Wagner, A., Jones, E.A.: Cognitive Tutor for Genetics Problem Solving: Learning Gains and Student Modeling. Journal of Educational Computing Research 42(2), 219–239 (2010)
9. D'Mello, S. K., Lehman, B., & Person, N. (2010). Monitoring affect states during effortful problem solving activities. International Journal of Artificial Intelligence in Education, 20(4), 361–389., doi:10.3233/JAI-2010-012
10. Girard, S., Chavez-Echeagary, H., Gonzalez-Sanchez, J., Hildalgo-Pontet, Y., Zhang, L., Burleson, W., and VanLehn, K., (2013), Defining the behavior of an affective learning companion in the affective meta-tutor project, in K. Yacef et al. (Eds.): Proceedings of the 16th international conference on Artificial Intelligence in EDucation (AIED'13), LNAI 7926, pp. 21--30. Springer-Verlag, Berlin, Heidelberg.
11. Gowda, S.M., Pardos, Z.A., and Baker, R. S. J. D. 2012. Content learning analysis using the moment-by-moment learning detector. In Proceedings of the 11th international conference on Intelligent Tutoring Systems (ITS'12), Stefano A. Cerri, William J. Clancey, Giorgos Papadourakis, and Kitty Panourgia (Eds.). Springer-Verlag, Berlin, Heidelberg, 434-443. DOI=10.1007/978-3-642-30950-2_56
12. Koedinger, K.R., Corbett, A.T., Perfetti, C. (2010): The Knowledge-Learning-Instruction (KLI) Framework: Toward Bridging the Science-Practice Chasm to Enhance Robust Student Learning. Carnegie Mellon University Technical Report, June, 2010
13. Martin, J., VanLehn, K.: Student assessment using Bayesian nets. International Journal of Human-Computer Studies 42, 575–591 (1995)
14. Lachaume, T., Girard, P., Guittet, L., & Fousse, A. (2012). ProtoTask, new task model simulator. In M. Winckler, P. Forbrig, & R. Bernhaupt (Eds.), Human-Centered Software Engineering (Vol. 7623, pp. 323– 330). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-642-34347-6
15. Muldner, K., Burleson, W., Van, D. S., Brett, & Vanlehn, K. (2011). An analysis of students' gaming behaviors in an intelligent tutoring system: predictors and impacts. User Modeling and User-Adapted Interaction, April 2011, 21(1-2), 99–135m doi:10.1007/s11257-010-9086-0
16. Shih, B., Koedinger, K.R., Scheines, R.: A response time model for bottom-out hints as worked examples. In: Proceedings of the 1st International Conference on Educational Data Mining, pp. 117–126 (2008)
17. Walonoski, J. A., & Heffernan, N. T. (2006). Prevention of off-task gaming behavior in intelligent tutoring systems Proceedings of the 8th international conference on Intelligent Tutoring Systems. Jhongli, Taiwan Berlin, Heidelberg.
18. Zhang, L., Burleson, W., Chavez-Echeagary, H., Girard, S., Gonzalez-Sanchez, J., Hildalgo-Pontet, Y., and VanLehn, K., (2013), Evaluation of a meta-tutor for constructing models of dynamic systems, in K. Yacef et al. (Eds.): AIED'13, LNAI 7926, pp. 666--669. Springer-Verlag, Berlin, Heidelberg.

Validating Item Response Theory Models in Simulated Environments

Manuel Hernando, Eduardo Guzmán and Ricardo Conejo

E.T.S. Informática. Universidad de Málaga,
Bulevar Louis Pasteur, 35. 29071
Málaga, Spain
{mhernando, guzman, conejo}@lcc.uma.es

Abstract. The Item Response Theory is a successful technique generally used in testing systems. Its application in problem solving environments requires the collection of large amount of data. That issue is stressed with ill-defined domains in which the actions that a student could accomplish are difficult to predict. Known IRT models could not be as appropriate as it is desired to that application and we have to explore new alternatives. One of these alternatives is a new family of models called quasipolytomous models of IRT. These models are halfway between dichotomous and polytomous models and require less data than polytomous models being more informative than dichotomous ones. Validating these new models is a very difficult issue in a real environment since student knowledge level is not observable. A simulation environment could help us to verify new models of IRT. Besides, with a simulator we can study different scenarios and observe how our model behaves in them.

Keywords: Problem Solving Environments, Student Modeling, Procedural Knowledge Estimate, Item Response Theory.

1 Introduction

Student modeling in problem solving environment is an important issue in the AIED field. Constraint-Based Modeling (CBM) [1] and Cognitive Tutors (CT) [2] are the outstanding approaches in that matter. CBM models are a set of constraints associated to principles in the domain that could be either violated or not, those constraints are related to declarative principles of the domain. CT inferred procedural student knowledge directly from student interactions through a technique called Knowledge Tracing [2] which is based on Bayesian procedures and estimates the probability that a student has learned a certain rule of the domain given his/her actions.

Procedural knowledge could be also inferred by other techniques such as the Item Response Theory (IRT) [3], which is one of the most important strategies of declarative knowledge assessment in testing systems. Our proposal of applying IRT to problem solving environment sets a connection between problem solving environment and testing, that is, if we make a matching between the elements of problem solving and the elements of testing we can, directly, apply IRT to infer procedural knowledge. In this sense, we model the solution of each problem as a directed graph where nodes are states of the solution path of the problem and edges are transitions between states. Using that representation each node could be understood as a question and each edge as an option in the question.

Our challenge is also to develop an automatic (or semiautomatic) procedure for mining the problem solving path from the logs of students' performance while solving it. This mining process would lead us as well to infer the IRT components which will be used for diagnosing the procedural knowledge. Furthermore, we also want this procedure to be dynamic, that is, the solution path and the inference of IRT components have to be updated dynamically when new logs will be available. Accordingly, the problem graph will include all possible student actions, so when a student completes an action that never was completed by another student it has to be included in the problem graph. These new actions have to be taken into account when the procedural knowledge is assessed, that is, the calibration procedure of the IRT has to be done when new actions are incorporated to the problem graph.

There are, mainly, two families of IRT-based models according to how they update the estimated student knowledge in terms of the student's response: dichotomous and polytomous models. Dichotomous models consider each response as either correct or incorrect whereas polytomous models consider each response individually. These traditional IRT models could not be as good as it is expected dealing with this type of calibrations since dichotomous models are not as informative as we need in this kind of problems and there will be actions with little evidence (maybe actions followed by 1 or 2 students) to polytomous calibration, especially in ill-defined domains where the set of possible actions is very large.

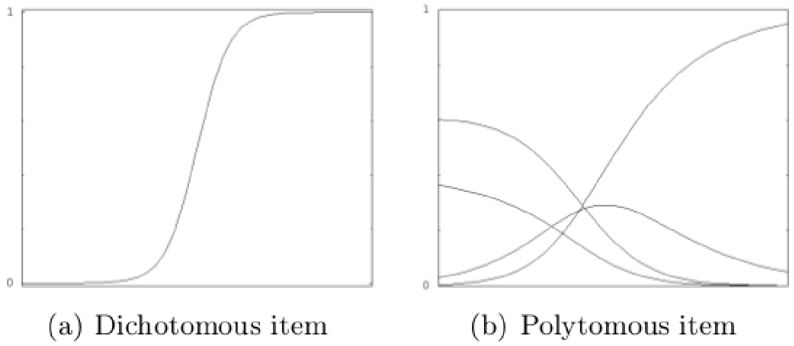
In order to explore new IRT models that fit better with our challenges we have developed a simulation environment in which virtual students (with a known real procedural knowledge of the domain) solve virtual problems (simulating their behavior according to their prior knowledge) and we have compared their estimated knowledge with their real knowledge. In this sense, we have developed a new family of IRT-based models called *quasipolytomous models* which are halfway of dichotomous and polytomous models considering not all possible responses but a subset of them.

2 Item Response Theory Models

The IRT is one of the most successful and well-founded strategies for knowledge inference in testing systems [3]. IRT infers and models student performance by means of some probabilistic functions called characteristic curves, the idea is that student's results could be explained by a set of non-observable factors (for instance, the knowledge level).

There are a lot of IRT models, based on how the models update the estimated student knowledge in terms of his/her response they could be [4]:

- Dichotomous models: Each response is considered as either correct or incorrect. When a student selects an option in a question test, his/her estimated knowledge is updated according to whether the option selected is the correct one or if that is other. These models require only a characteristic curve per item that represents the probability that a student with a certain knowledge level answers it correctly. This characteristic curve is called item characteristic curve (ICC).
- Polytomous models: Each possible response has its own characteristic curve called operating characteristic curve (OCC) [5], which expresses the probability that a student select that answer [6]. The student estimated knowledge is updated by means of the OCC related to the selected response.



(a) Dichotomous item

(b) Polytomous item

Fig. 1.Characteristic curves of an item under dichotomous and polytomous models

Polytomous models are more informative than dichotomous ones since they take into account each possible response independently instead of considering each answer as correct or incorrect.

Figure 1 shows the curves of an item of both the dichotomous and polytomous models of IRT. The dichotomous model is shown in Figure 1(a) and has only the ICC which is the probability of answering correctly this item (y-axis), given a certain knowledge level (x-axis). The polytomous model is presented in Figure 1(b).Each item choice has its own characteristic curve which is the probability of choosing this choice (y-axis), giving a certain knowledge level (x-axis).

The most popular proposals for modeling the dichotomous characteristic curves are the logistic models, which use logistic functions. These models could be classified, considering the number of parameters that the function has. According to this classification there are 3 kinds of logistic models: 1PL, 2PL, and 3PL, with one, two and three parameters, respectively. A generic 3PL ICC of an item X_i is defined as follows:

$$P(X_i|\theta) = c_i + (1 - c_i)\frac{1}{1+e^{-Da_i(\theta - b_i)}} \tag{1}$$

Where D is a parameter introduced to fit the curve similar to the normal curve, parameter a_i is the discrimination, parameter b_i is the difficulty, parameter c_i is the guessing parameters of the item X_i, and θ is the knowledge level.

There are a lot of polytomous models of IRT. In our work we have considered the IRT-model proposed by Thissen and Steinberg for multiple-choice items [7]. In this model besides the observable categories (selectable choices) there is another non-observable and latent category called "*don't know*" (DK) that expresses the probability that a student does not know how to answer the item. Each observable category has a portion of the category DK included since students who do not know will select an observable category. The formula of each observable category is exposed below, X_i represents the response to the item i and h is the category selected:

$$P(X_i = h|\theta) = \frac{e^{a_h\theta+c_h}}{\sum_{k=0}^{m_i} e^{a_k\theta+c_k}} + d_h \frac{e^{a_0\theta+c_0}}{\sum_{k=0}^{m_i} e^{a_k\theta+c_k}} \quad (2)$$

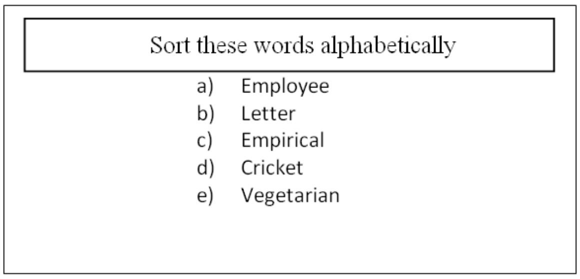
category 0 is the non-observable category DK and d_h is the portion of that category included in each observable category h. The parameters denoted by a reflects the order, as well as discrimination, for the categories, and the parameters denoted by c reflect the relative frequency of the selection of each alternative.

3. Introducing IRT in problem solving

The application of IRT to problem solving environments requires a polytomous model since each student action should be taken into account; a dichotomous model only would be able to establish if an action is correct or incorrect. However, some actions could have little evidence and IRT calibration could be not as accurate at it is expected. For that reason, we have developed a new family of models of IRT called quasipolytomous models of IRT which are on the halfway between dichotomous and polytomous ones.

Quasipolytomous models consider not all choices as independent but only those that have enough evidence. For instance, let us consider an item witch 20 choices (what is usual in problem solving environment if we include all student actions), if 8 of them have been selected only by 1 or 2 students they do not offer us enough evidence to do a polytomous calibration. Instead of doing it, we consider these 8 choices as a simple choice reducing the number of OCCs from 20 (one per choice) to 13 (12 individual choices and an extra choice that group the other 8) that have, all of them, evidence enough to do an IRT calibration.

In testing systems there could be items with a lot of choices too, let us consider figure 2 in which the number of choices is 120 since we have to take into account the permutation between these 5 elements.



Sort these words alphabetically

a) Employee
b) Letter
c) Empirical
d) Cricket
e) Vegetarian

Fig. 2. An item with 120 choices

3 Simulation environment

In order to verify quasipolytomous models we have developed a simulator in which virtual students have a real (prior) knowledge assigned and they have to solve some virtual problems according to their knowledge level. Once the students solve the proposed problems, their knowledge is estimated by means of a quasipolytomous model of IRT and then, these estimations are compared with the students' real knowledge.

3.1 Virtual problems

In this simulation environment, a virtual problem is represented as a collection of items; each item is a state of problem solving path with a certain number of possible transitions to other states. These transitions are the choices of the item.

While students are solving problems, new students' actions could appear in the system and they need to be included in the model. For that reason, virtual problems are not static entities but they can change during students' interactions.

At the beginning, each problem has only the ideal solution path, that is, those nodes that are part of the ideal solution. Those nodes are modeled by dichotomous items according to the equation 1.A characteristic curve, the ICC, is enough to model this kind of items. Students who do not answer correctly the item could, with some probability, make a new action at this step of problem solving. Then the opposite curve (i.e. 1-ICC) is branched into two curves changing the original dichotomous item to a polytomous one. According to the former explanation, students' actions could provoke the addition of new nodes to the problem graph.

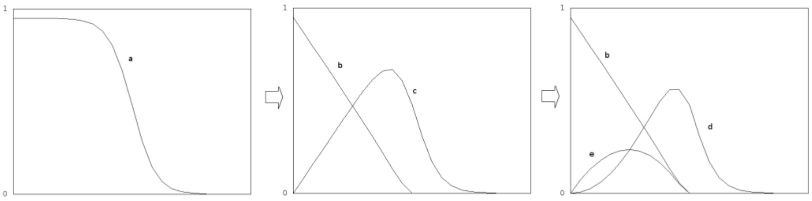


Fig. 3. Addition of new curves to an item

Figure 3 shows how new curves are added to an item. Firstly we have the opposite ICC curve which expresses the probability that a student, given his/her knowledge level, will answer the item incorrectly. This is curve marked as *a* in the figure. Curve *a* is branched into curves *b* and *c* when the action represented by curve *c* is added to the model; finally, curve *c* is branched again and then curves *d* and *e* are added to the item. The item was, at the beginning, a dichotomous item with two possible responses; then a new action was included in it and, as a consequence, the wrong curve was converted in other two curves. Again a new action was added to the model and the item changes to a 4-choices item.

3.2 Virtual students

A virtual student is an entity that has a real (prior) knowledge associated and is able to solve problems according to it. The idea is that, depending on its knowledge level, the student could go on through the problem graph by completing an action or other. Further, they could accomplish new actions with a certain probability.

A student will select an action of the state according to his/her knowledge level and the characteristic curve since these curves are probabilistic functions of the knowledge level. Figure 4 shows an example of a virtual student selecting a choice in a 4-choice item. That student will select each choice with a probability of 0.2, 0.38, 0.18 and. 0.24 respectively. When a non-correct action is selected, there will be a probability that the student will accomplish a new action and, therefore, this curve will be branched into other two.

Since real knowledge of virtual students is assigned before simulation, we can choose what population distribution we want in each experiment. It is an important feature of the simulation since we can study the impact of different kind of population in order to validate new models in all cases or only in some of them.

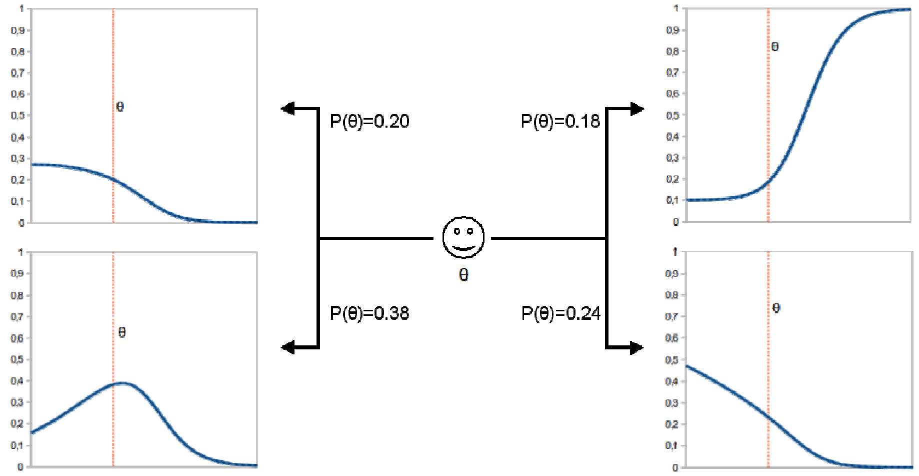


Fig. 4. Virtual student selecting a choice

4 Experimentation

In order to verify our model in a simulated environment, we have conducted some experiments with virtual students and virtual problems. The experiments have been accomplished to verify the model with different population distributions as well as different models of IRT and different number of problems.

4.1 Experiment description

The experiments accomplished with our simulator were able to compare scores offered by quasipolytomous models of IRT in different situations. Each experiment has been accomplished 30 times in order to reduce the impact of anomalous data. The number of virtual students in each experiment was 1000.

In order to get confidence results we have not done the calibration IRT phase, instead of that, we have estimated knowledge level with the known curves of each model. Polytomous models could not be as accurate as expected including the calibration stage since some item choices could have not enough evidence to be calibrated properly. The student estimated knowledge level is calculated using the formula of the equation 3, the probability of having a specific knowledge level given the steps followed by the student solving the problem is calculated multiplying the probability of selecting each step given the knowledge level.

$$P(\theta|s_1, s_2, \cdots, s_n) = \prod_{i=1}^{n} P(s_i\,|\theta) \tag{3}$$

In our experiments, virtual students have solved two problems with 10 items. We have accomplished mainly three experiments varying the population distribution, the probability of generating new actions and the item difficulty respectively. In our experiments, we have compared accuracy of knowledge level estimation obtained by a quasipolytomous model with those obtained by polytomous and dichotomous models with the same data.

The accuracy of knowledge level estimation was calculated using the next formula, where θ is the real knowledge, θ^* is the estimated knowledge, and N the number of students:

$$d(\theta,\, \theta^*) \;=\; \frac{\sum_{i=1}^{N}(\theta-\theta^*)^2}{N} \tag{4}$$

We have chosen four types of population distribution to accomplish the experiment. First, a normal distribution of population was selected; the probability of having a knowledge level was centering in the middle of knowledge range, since that range was [-3, 3]; the distribution was centered in 0. The second population was a uniform one, in which the probability of having a knowledge level is equally distributed. The other two populations were a low-level and a high-level population, which were normal distributions centered in small and high values respectively. Figure 5 shows the knowledge level distributions used in our experiments. From left to right they are the normal, the uniform, the low-level, and the high-level distribution respectively.

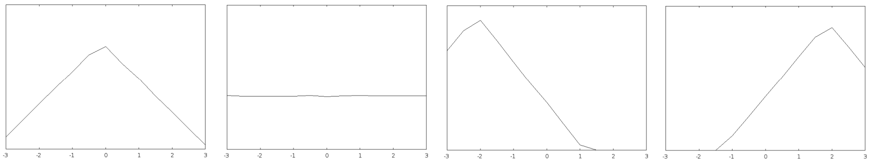


Fig. 5. Students' knowledge level distributions used in the experiments

The impact of the generation of new actions was also studied in our experiments. To this end, we have conducted experiments varying the percentage of adding a new action giving it values of 1%, 1.5%, 2%, and 2.5%. More than 2.5% of adding new actions could lead to a very large number of curves in the model.

Finally, we have also considered the difficulty of the items. We have conducted an experiment changing the value of this parameter. The difficulty value is calculated according to a normal distribution centered in a certain value, which is the difficulty average. We have done our experiments with values of -1, 0, and 1, respectively.

4.2 Experiment results

Experiments conducted suggest that polytomous models of IRT perform more accurately than dichotomous ones. This result is not surprising, since polytomous models are more informative. Experiments also show that quasipolytomous models of IRT are not as accurate as polytomous ones but more accurate than dichotomous ones. Besides, results obtained by quasipolytomous models are very similar to those obtained by polytomous ones.

Table 1 shows results obtained with different population distribution. In all cases accuracy obtained by polytomous models are better than obtained by dichotomous and quasipolytomous models. These differences are higher in the normal distribution and lower with a low-level population.

Table 1. Accuracy of IRT models changing the population distribution

Distribution	dichotomous	quasipolytomous	polytomous
Normal	4.153118	0.7957639	0.6714305
Uniform	3.841433	0.6284451	0.5397757
Low-level	2.290098	0.4186174	0.2723583
High-level	3.784738	0.8671618	0.8108576

We can see in the former table that differences between results obtained by quasipolytomous models and polytomous ones are not significant.

Our second experiment conducted studied how affects the probability of adding a new action. To this end, we changed the percentage of adding a new action from 1%

to 2.5%. Table 2 shows those results, when we increase the probability of adding a new action the accuracy of dichotomous and quasipolytomous models gets worse since but the accuracy of the polytomous models gets better since they obtain more precise information.

Table 2. Accuracy of IRT models changing the percentage of adding a new action

% new action	dichotomous	quasipolytomous	polytomous
1.0%	3.291380	0.6609889	0.6126646
1.5%	3.489956	0.6695576	0.5845097
2.0%	3.613464	0.6778118	0.5593333
2.5%	3.674588	0.7016298	0.5379146

Finally, we have compared results obtained by these models varying item (solving path step) difficulty average. The student knowledge level used in our experiment is in the range [-3, 3]. We have considered average item difficulty of -1, 0, and 1. An item with difficulty of b_i will be answered correctly by a student with knowledge level of b_i with a probability of 0.5. Table 3 shows the results of this experiment, the dichotomous models have a better behavior when the difficulty is below 0, however quasipolytomous and polytomous models get better results when the difficulty increases. Polytomous and quasipolytomous models accurate better when the item difficulty is higher since it allows the model to have more curves to estimate the student knowledge level.

Table 3. Accuracy of IRT models changing the item difficulty average

Difficulty	dichotomous	quasipolytomous	polytomous
b_i= **-1**	2.364223	1.2011291	1.0558448
b_i **= 0**	3.131082	0.5410807	0.4450589
b_i **= 1**	5.056735	0.2902812	0.2199130

5 Conclusion

In this paper, we have validated a new approach that uses the Item Response Theory, a well-founded theory generally used for declarative knowledge estimation in testing systems, to infer procedural skills in problem solving environments. To do that, we have developed a new model of IRT, the quasipolytomous model. This model is halfway between dichotomous and polytomous models being more informative than dichotomous models and needing less amount of data than polytomous ones.

This verification could be difficulty accomplished in a real environment, since we need to know the prior knowledge level of students to measure the estimation accuracy. This knowledge level, however, is a latent trait that is not observable. In addition, we needed a controlled environment where the students' performance was not biased by external factors. For all these reasons we have developed a simulation environment.

Using a simulation environment we can choose the nature of the population in order to study how well the model performs in different situations and with different students' samples. We also can decide the difficulty of the steps of the problem (i.e. items): before any calibration, the simulator is able to decide what items are more difficult and what are easier.

Other advantage of using a simulation environment is that we can repeat each experiment in order to reduce the impact of anomalous data. Furthermore, we can present the problems to a large number of students, which could be difficult in a real environment.

Regarding the quasipolytomous model of IRT, our experiments show that its application is useful in problem solving environment (besides any kind of procedural task, and in declarative domains for inferring declarative knowledge), especially if we work with ill-defined domains in which the amount of possible new actions is very large. Quasipolytomous models of IRT offer similar estimations as polytomous models but need less data. Besides, quasipolytomous models of IRT are more informative than dichotomous ones since they collect data from correct and incorrect responses.

Acknowledgements: This work is part of DEDALO project which is financed by the Andalusian Regional Ministry of Science, Innovation and Enterprise (P09-TIC-5105).

6 References

1. Mitrovic, A., Martin, B., Suraweera, P.: Intelligent tutors for all: The constraint-based approach. IEEE Intelligent Systems 22 (2007) 38-45
2. Anderson, J.R., Corbett, A.T., Koedinger, K.R., Pelletier, R.: Cognitive Tutors: Lessons learned. The journal of the Learning Sciences 4 (1995) 167-207
3. Embretson, S.E., Reise, S.P.: Item response theory for psychologists. Lawrence Erlbaum, Mahwah (2000)
4. Guzmán, E., Conejo, R., de-la Cruz, J.L.P.: Adaptive testing for hierarchical student models. User Model. User-Adapt. Interact. 17 (2007) 119-157
5. Dodd, B.G., De Ayala, R.J., Koch, W.R.: Computerized adaptive testing with polytomous items. Applied Psychological Measurement 19 (1995) 5-22
6. Guzmán, E., Conejo, R.: A model for student knowledge diagnosis through adaptive testing. In: In Proceedings of 7th International Conference Intelligent Tutoring Systems, ITS2004 Brazil, Springer-Verlag (2004) 12-21
7. Thissen, D., Steinberg, L.: A response model for multiple choice items. Psychometrika 49 (1984) 501-519

Toward a reflective SimStudent: Using experience to avoid generalization errors

Christopher J. MacLellan, Noboru Matsuda, and Kenneth R. Koedinger

Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh PA 15213, USA
cmaclell@cs.cmu.edu, mazda@cs.cmu.edu, and koedinger@cmu.edu

Abstract. Simulated learner systems are used for many purposes ranging from computational models of learning to teachable agents. To support these varying applications, some simulated learner systems have relied heavily on machine learning to achieve the necessary generality. However, these efforts have resulted in simulated learners that sometimes make generalization errors that the humans they model never make. In this paper, we discuss an approach to reducing these kinds of generalization errors by having the simulated learner system reflect before acting. During these reflections, the system uses background knowledge to recognize implausible actions as incorrect without having to receive external feedback. The result of this metacognitive approach is a system that avoids implausible errors and requires less instruction. We discuss this approach in the context of SimStudent, a computational model of human learning that acquires a production rule model from demonstrations.

Keywords: simulated learners, metacognition, cognitive modeling, representation learning, grammar induction, generalization error

1 Introduction

Simulated learning systems can be used for a wide range of tasks, such as modeling how humans learn, as teachable agents, and as a means to automate the construction of models that can be used in cognitive tutors. In an effort to reduce the amount of developer effort needed to deploy simulated learners for these tasks, researchers have been relying increasingly on the use of machine learning algorithms. However, by increasing the generality of these systems through machine learning approaches, these systems become more susceptible to making unrealistic generalization errors.

When using simulated learners to model human learning, we desire systems that predict student's errors as well as their correct behavior. Unrealistic generalization errors, in the context of these systems, are errors that the system predicts humans will make, but that they never actually make. If a system is prone to making these kinds of errors, then it becomes difficult to draw conclusions from the predictions the simulated learners makes for novel tasks.

These generalization errors also complicate the use of simulated learners as teachable agents because they result in a system that produces non-human behavior. When human students are teaching a simulated learner in a peer-tutoring scenario and it makes errors that humans never make, then it decreases the authenticity of the experience. This inauthenticity might effect the social dynamics of the learning-by-teaching scenario possibly making the teachable agent less effective.

Finally, generalization errors also have negative effects when using simulated learners to automatically build cognitive tutors. For this purpose, simulated learners have been used to author production rule models via interactive demonstrations of the solutions to the problems the system will tutor. This approach may decrease the amount of work required to build a cognitive tutor and allow subject-matter experts to author tutors directly, without an AI developer. In this paradigm, SimStudent's errors are useful to the extent that they correspond with typical student errors; in these cases, the resulting production rules can be added to the tutor's bug library. However, if the errors are unrealistic, the author must waste time identifying and deleting these nonsensical production rules.

In this paper, we propose an approach that uses background knowledge to mitigate unrealistic generalization errors with no changes to the underlying algorithms and which should increase the effectiveness of the underlying learning mechanisms. Before presenting this approach in section 4, we first review SimStudent, the simulated learning system that provides the context for this work (section 2) and introduce a motivating example of a nonsensical generalization error SimStudent currently makes (section 3). After presenting this approach, we present some initial results and discuss conclusions and future work.

2 The SimStudent Architecture

The simulated learner system that we focus on in this paper is SimStudent, a system that induces production rule models from demonstration and problem solving. The SimStudent system is used primarily for three tasks: to model and predict human learning, to author cognitive tutors, and to function as a teachable peer-agent.

In order to understand how SimStudent works and the situations in which it makes unrealistic generalization errors, we will review the types of knowledge used by SimStudent, how this knowledge is represented, and the learning mechanisms SimStudent uses to acquire this knowledge from experience.

2.1 Knowledge and Representation

There are three kinds of knowledge in SimStudent: primitive operator function knowledge, conceptual knowledge, and procedural knowledge. The first kind of knowledge is hand-constructed and consists of the low-level functions for manipulating data available to the system (i.e., adding two values, appending two strings together, etc.). One example of a low-level function is SkillAdd, which accepts two arguments, each of type arithmetic expression, and returns the sum of these two expressions as a single arithmetic expression. These functions constitute SimStudent’s background knowledge. Depending on the task SimStudent is being used for, different kinds of background knowledge may be appropriate.

Head	**Body**	**Prob**
Expression	← Number Variable	0.95
Expression	← Minus Variable	0.05
Variable	← x	1.0
Minus	← -	1.0
Number	← 0	0.1
	...	
Number	← 9	0.1

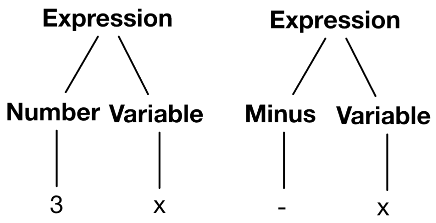


Fig. 1. A simple probabilistic context-free grammar and example parses of two expressions using this grammar.

The second kind of knowledge is conceptual, or representational, knowledge, which is encoded as a probabilistic context-free grammar. It is automatically acquired by SimStudent and is used to interpret the interface and information in it. Figure 1 shows a simple example of the conceptual knowledge SimStudent might possess about expressions for an algebra domain. This knowledge enables SimStudent to automatically extract plausible “chunks” from the input, such as the coefficient or term in an equation, which can subsequently be manipulated by primitive operator functions or procedural rules. Furthermore, this knowledge can be used to determine the likelihood that a given example was produced by the grammar.

If (current-row ’output-cell ’row) **then** (write-text ’output-cell
(cell-in-row ’row 1 ’left-side) → (append “divide” ’coefficient)).
(is-left-child-of ’left-side ’coefficient)

Fig. 2. An example production rule for division.

The final kind of knowledge is procedural knowledge, which represents the skills that we desire students to learn. This knowledge is encoded as production rules, which contain conditions under which the rules apply and what to do under those conditions. Figure 2 shows an example of a production rule signifying that when the left side of the equation’s parse tree has a left child (here called coefficient), then enter “divide <the coefficient>” into the output cell.

2.2 Learning Mechanisms

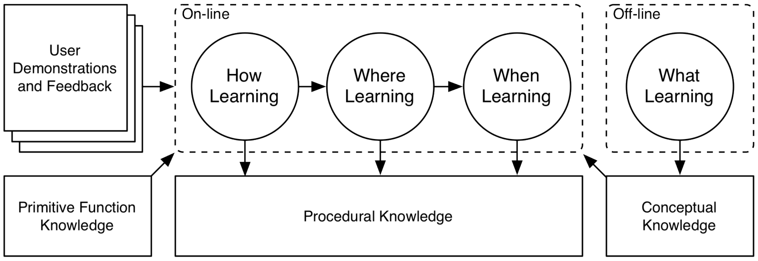


Fig. 3. A diagram of the SimStudent learning mechanisms and how they interact.

Of the three kinds of knowledge manipulated by the SimStudent system, two are learned automatically: the conceptual and procedural knowledge. To acquire these two kinds of knowledge the system employs four learning mechanisms: what learning, where learning, when learning, and how learning. The what learning is used to acquire the conceptual knowledge whereas the where, when, and how learning are used to acquire the procedural knowledge. Figure 3 shows how these four learning mechanisms interact. Before SimStudent is used, the what learning is run to acquire the conceptual knowledge. When SimStudent encounters a situation where it does not know how to act, which is common initially, it requests a demonstration from the author (the tutor developer or student tutor). This demonstration is comprised of four parts:

- *Focus of attention:* the set of relevant interface elements (e.g., the left and right hand sides of an equation);
- *Selection:* the interface element to manipulate (e.g., the output cell);
- *Action:* the action taken in the selection (e.g., update the text value); and,
- *Input:* the argument to the action (e.g, the text string used to update the selection).

Every time the system sees a new demonstration or gets corrective feedback on its performance, it learns or modifies a production rule. Production rule learning is done in three parts: 1) how learning attempts to explain the demonstration and produce the shortest sequence of primitive operator functions that replicates the demonstrated steps and ones like it, 2) where learning identifies a generalized path to relevant elements in the tutor interface that can be used as arguments to the function sequence, and 3) when learning identifies the conditions under which the learned production rule produces correct actions. We will now review each of these learning mechanisms.

What This mechanism operates off-line to acquire a probabilistic context-free grammar from only positive examples. This task can be defined as:

- *Given:* a set of examples of correct input;
- *Find:* a probabilistic context-free grammar with the maximal likelihood of producing the examples.

This task is performed using a grammar induction approach outlined by Li et al. [1], which uses a greedy approach to hypothesize the grammar structure and Expectation Maximization to estimate the grammar parameters.

Whenever a demonstration is given to SimStudent, it augments the provided information with the most likely parse trees of the content of each element in the focus of attention. This additional information is used by SimStudent in the subsequent learning mechanisms to extract deep feature knowledge from the content (e.g., to recognize and extract the coefficient of a term in an equation). The parse trees make this deep feature information directly accessible to SimStudent through the nodes in the parse tree (e.g., the left child of the parse tree for "$3x$" in Figure 1 corresponds to the coefficient).

How This is the first of three mechanisms executed in response to a demonstration. The how learning task can be defined as:

- *Given:* a set of demonstrations consisting of the state of the relevant interface elements and the parse trees of the contents of these elements as well as the resulting input for each state;
- *Find:* a sequence of primitive functions that when applied to each state produces the corresponding input.

This task is performed by exhaustively applying the primitive operator functions over all nodes in the focus of attention parse trees until the input is produced. The iterative-deepening depth-first search strategy is used to find the shortest sequence of functions that explains the data [1]. If no sequence exists, then a special functions is created that takes the states and produces the corresponding inputs.

Where This learning mechanism identifies the path to the relevant tutor interface elements. The tutor interface elements are specified by a hierarchical tree structure (a table is comprised of rows which each contain cells). During interactive instruction, the relevant interface elements are specified by the author teaching SimStudent. For each relevant element, SimStudent generates a parse tree for the contents. The relevant portions of these parse trees are defined as those that are utilized by the operator function sequence acquired through the how learning. The task of learning a general path to this relevant information can be defined as:

- *Given:* a hierarchical representations of the interface elements and their parse trees, the function sequence from the how learner, and a set of elements that have been identified as relevant;

- *Find:* a list of paths through the representation hierarchy to all of the relevant elements and the relevant portions of their parse trees.

The SimStudent approach to this task is to conduct specific-to-general learning over the set of relevant interface elements and parse trees [1]. Returning to the table examples, if the first cell in the first row of the table is always relevant, then a path to that specific cell will be returned. However, if all of the elements in the row are specified as relevant, then the entire row will be returned. After the location to the relevant elements has been identified, the system utilizes the function sequence to identify the relevant portions of the parse trees for each element. This same specific-to-general learning is then conducted over these relevant parse trees (within each element).

When This final mechanism identifies the conditions when the learned production rule is applicable. This task is defined as:

- *Given:* a set of positive and negative examples, each consisting of a set of features and their associated label;
- *Find:* a set of conditions over the features that separate the positive and negative examples.

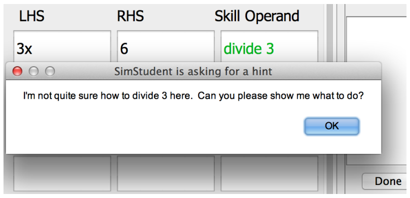
As specified, this is a supervised learning task. The features used by SimStudent to represent each example are predicates that are automatically generated from the relevant portions of the parse trees. For example, there exists an "is-left-child-of" predicate, which says that a particular argument is the left child of a given node in one of the parse trees. This type of feature enables the retrieval of equations, terms, coefficients, and variables. Given the feature descriptions of each example, the positive and negative labels come from the user instructing the SimStudent system. The first positive example is the initial demonstration. Subsequent examples are generated when SimStudent tries to use the learned rules to solve novel problems and receives yes/no feedback from the author. To derive the set of conditions given the examples, SimStudent uses the FOIL algorithm [2], which uses information theory to perform a general-to-specific exploration of the space of hypothetical conditions.

These four learning mechanisms result in a simulated learning system that accepts user demonstrations and feedback and automatically acquires probabilistic context-free grammar rules and production rules. The system requires little background knowledge; for each task only the primitive functions need to be defined by the developer. However, the cost of this generality is a system that sometimes makes unrealistic generalization errors.

3 An example of an unrealistic generalization error

To explore the types of generalization errors that SimStudent makes, we turn to the algebra domain. One of the skills that students learn in this domain is how to proceed when given a problem of the form $< Symbol >< Variable >=<$

Symbol > (e.g., $3x = 6$). The skill that we desire the student to learn in this situation is to specify that their next step is to divide both sides by the coefficient of the term on the left side of the equation (the production rule from Figure 2).



LHS	RHS	Skill Operand
3x	6	divide 3

SimStudent is asking for a hint

I'm not quite sure how to divide 3 here. Can you please show me what to do?

Fig. 4. SimStudent requesting a demonstration in an algebra tutor interface after the author has just entered "divide 3."

When SimStudent is first presented with a problem of this form, such as $3x = 6$, it will inform the author that it does not know how to proceed and ask for a demonstration. The author might demonstrate to SimStudent that the cells containing the left and right hand sides of the equation are relevant to the problem (by double-clicking on these cells) and update the next step interface element with "divide 3" (see Figure 4).

After receiving this demonstration, SimStudent parses the contents of the focus of attention (The first parse tree in Figure 1 shows an example of what the left hand of the equation might look like). Next, it employs the how learning mechanism, which searches for a sequence of functions that when applied to the nodes in the parse tree produce the input. In this example, it might learn to append the left child of the parse tree (for the left side of the equation) to the word "divide" and place it into the tutor interface (the then part of the production rule in Figure 2). Using the locations of the relevant elements (the left child of the parse tree), SimStudent then learns a general path through the representation hierarchy to the relevant elements and the relevant portions of the parse trees for these elements. Finally, SimStudent runs FOIL over the relevant information to learn the conditions under which the learned behavior is applicable. This results in the if portion of the production rule in Figure 2.

The learned production rule is more general than the single demonstration it was learned from; it is applicable for many equations, such as $4x = 12$ or $2x = 8$. However, when SimStudent is presented with a subtly different example that utilizes the same skill, $-x = 2$, it results in the mistaken generation of the input "divide -" (instead of "divide -1"). This is because in this situation the left child of the parse tree on the left hand side of the equation is a minus sign instead of the coefficient (see the second parse tree in Figure 1). In a review of problems of the form $-x =< Constant >$ in the 'Self Explanation CWCTC Winter 2008 (CL)' dataset accessed via DataShop [3], none of the human student made this error– therefore it is an example of unrealistic generalization error.

4 Reflecting before Acting

One reason that humans do not make this error is that they have a "sense" for what are reasonable output actions and they (subconsciously) reflect on actions before taking them. When a student is faced with the problem $-x = 2$ they may mentally produce the output "divide -," but realize that a "-" by itself is not mathematically grammatical because they have never seen an instance where this has occurred. This might lead them to consider a different action or to ask for help.

To reproduce this type of behavior, we modified SimStudent to utilize its conceptual knowledge, the probabilistic context-free grammar trained on example inputs (described as "what" learning in section 2). The acquired grammar is used to recognize when a potential output is not grammatical (when it cannot be parsed) and automatically flag the situation as a negative example. In other words, the system supervises itself and provides negative feedback (which the when learner uses) to improve its learning.

Now, when SimStudent is presented with a problem and finds an applicable rule, it simulates the execution of the rule and constructs a probabilistic parse of the value generated by the rule. If the value cannot be parsed by the current grammar (there is a 0% probability that the grammar produced the value), then SimStudent flags the trace as a negative instance and re-runs the when learning, which refines the conditions of the rule so that it no longer applies in the erroneous situation. If SimStudent has no other applicable rules, then it request a demonstration from the author, exactly like a human student.

5 Initial Results

To evaluate the effectiveness of this metacognitive loop, we have tested the probabilistic parser's ability to separate correct from incorrect actions based on the parse probability defined by the probabilistic context-free grammar. Table 1 shows five problems where SimStudent might make unrealistic errors. The first three are problems where SimStudent might induce a rule for dividing by the symbol before the variable instead of the coefficient. The last two problems correspond to inducing a rule retrieving the symbol after the variable and division sign instead of the entire denominator. On all five problems, the probabilistic grammar was capable of identifying the correct from the incorrect actions.

These results suggest that this approach is capable of identifying these kinds of errors. In general, this approach will be effective at identifying errors that result in non-grammatical output, where grammatical is defined by the probabilistic context-free grammar. This is effective because the rules are learned specific-to-general on a substantial amount of positive example inputs. By bringing this previous experience to bare, SimStudent can avoid nonsensical generalization errors and produce its own negative feedback, which enhances the effectiveness of its other learning mechanisms (more self-labeled examples for the when learning). Furthermore, this requires no additional work from an author and should reduce the amount of required author feedback.

Table 1. Five examples of problems where SimStudent might make the generalization error of retrieving the character before the variable or after the variable and the division sign, the corresponding correct and incorrect actions, the validity of these actions, and the parse probability of the actions.

Example	Possible Action	Valid	Parse Probability
$-x = 2$	divide $-$	No	0.00%
	divide -1	Yes	19.64%
$(-2)x = 6$	divide)	No	0.00%
	divide (-2)	Yes	0.09%
$3(x + 1) = 6$	divide (	No	0.00%
	divide 3	Yes	27.90%
$x/(-3) = 3$	multiply (	No	0.00%
	multiply (-3)	Yes	0.09%
$x/-5 = 1$	multiply $-$	No	0.00%
	multiply -5	Yes	19.64%

This task of verifying the output could alternatively be viewed as applying constraints to SimStudent's output and learning from constraint violations. Viewed this way, our work is related to the work on constraint-based tutoring systems [4]. In our case, there is only one constraint, "the output must be grammatical" where grammatical is defined as the probability of the output being produced by the grammar must be greater than 0%. We use a threshold of greater than 0% to signify grammatical, but one could imagine using a different threshold (e.g., greater than 0.05%). Thus, this constraint could be viewed as a probabilistic constraint that is automatically acquired from positive training examples.

6 Conclusion and Future work

In this paper, we outlined a novel approach to detecting and learning from unrealistic generalization errors that can be employed by simulated learner systems. The implications of this approach are threefold: (1) its use will result in models of learning that more closely aligns with human data, (2) teachable agents using this approach will be more realistic for the students using them, and (3) developers can produce cognitive tutor models with less work.

While this approach shows promise, it clearly has some shortcomings that should be remedied in future work. First, a more in-depth analysis of the alignment between SimStudent and human students is necessary. Previous work [5, 6] has looked at the human errors that SimStudent is capable of predicting, but a more detailed analysis of the unrealistic generalization errors, or errors that SimStudent makes that human students do not, would be useful. This would serve as a baseline to evaluate the SimStudent model and to evaluate the effectiveness of this approach.

A second direction for future work is to compare this approach to other approaches that might reduce these errors. We could imagine a system that has additional condition knowledge for the operator functions so that it would not generalize to situations where the function sequence would not be applicable (such as trying to divide by a symbol instead of a number). It would also be interesting to explore how reflection might facilitate the acquisition of this additional condition knowledge for the operator functions.

Finally, we are interested in applying this approach in other more complex and open-ended domains such as in RumbleBlocks, an educational game that teaches K-3 children about the relationships between the concepts of stability, low center of mass, wide base, and symmetry. We have been exploring how probabilistic grammars can be used to learn conceptual knowledge in RumbleBlocks [7] and we believe that this approach should scale up to this more complex domain.

References

1. Li, N., Schreiber, A.J., Cohen, W.W., Koedinger, K.R.: Efficient Complex Skill Acquisition Through Representation Learning. Advances in intelligent tutoring systems **2** (2012) 149–166
2. Quinlan, J.R.: Learning Logical Definitions from Relations. Machine Learning **5** (1990) 239–266
3. Koedinger, K.R., Baker, R.S.J.d., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J.: A Data Repository for the EDM community: The PSLC DataShop. In Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.S.J.d., eds.: Handbook of Educational Data Mining. CRC Press (2010)
4. Mitrovic, A., Ohlsson, S.: Evaluation of a constraint-based tutor for a database language. International journal of artificial intelligence in Education **10** (1999) 238–256
5. Lee, A., Cohen, W.W., Koedinger, K.R.: A Computational Model of How Learner Errors Arise from Weak Prior Knowledge. In Taatgen, N., van Rijn, H., eds.: Proceedings of the Annual Conference of the Cognitive Science Society, Austin, TX (2009) 1288–1293
6. Matsuda, N., Cohen, W., Sewall, J., Lacerda, G., Koedinger, K.R.: Evaluating a Simulated Student using Real Students Data for Training and Testing. In Conati, C., McCoy, K., Paliouras, G., eds.: Proceedings of the International Conference on User Modeling. (2007) 107–116
7. Harpstead, E., MacLellan, C., Koedinger, K.R., Aleven, V., Dow, S.P., Myers, B.: Investigating the Solution Space of an Open-Ended Educational Game Using Conceptual Feature Extraction. In: Proceedings of the International Conference on Educational Data Mining. (2013)

Towards Moment of Learning Accuracy

Zachary A. Pardos† and Michael V. Yudelson‡

†Massachusetts Institute of Technology
77 Massachusetts Ave., Cambridge, MA 02139
‡Carnegie Learning, Inc.
437 Grant St., Pittsburgh, PA 15219, USA
zp@csail.mit.edu,yudelson@carnegielearning.com

1 Introduction

Models of student knowledge have occupied a significant portion of the literature in the area of Educational Data Mining[1]. In the context of Intelligent Tutoring Systems, these models are designed for the purpose of improving prediction of student knowledge and improving prediction of skill mastery. New models or model modifications need to be justified by marked improvement in evaluation results compared to prior-art. The standard evaluation has been to forecast student responses with an N-fold student level cross-validation and compare the results of prediction to the prior-art model using a chosen error or accuracy metric. The hypothesis of this often employed methodology is that improved performance prediction, given a chosen evaluation metric, translates to improved knowledge and mastery prediction. Since knowledge is a latent, the estimation of knowledge cannot be validated directly. If knowledge were directly observable, would we find that models with better prediction of performance also estimate knowledge more accurately? Which evaluation metrics of performance would best correlate with improvements in knowledge estimation? In this paper we investigate the relationship between performance prediction and knowledge estimation with a series of simulation studies. The studies allow for observation of the ground truth knowledge states of simulated students. With this information we correlate the accuracy of estimating the moment of learning (mastery) with a host of error metrics calculated based on performance.

2 Bayesian Knowledge Tracing

Among the various models of knowledge, a model called Bayesian Knowledge Tracing [2] has been a central focus among many investigators. The focus on this model has been in part motivated by its use in practice in the Cognitive Tutors [4], used by over 600,000 students, and by its grounding in widely adopted cognitive science frameworks for knowledge acquisition. For our experiments we will be employing the most frequently used basic Bayesian Knowledge Tracing

[1] A session during the main proceedings of EDM 2012 was dedicated to papers on Knowledge Tracing, a frequently used approach to modeling student knowledge.

model for both simulation and evaluation; however, there are implications beyond BKT models. Knowledge Tracing is a simple Hidden Markov Model of Knowledge defined by four parameters; two performance parameters and two knowledge parameters. The performance parameters, guess and slip, are the emission parameters in an HMM which respectively correspond to the probability that a student answers correct even if she is in the negative knowledge state (guess) and the probability that she answers incorrectly even if she is in the positive knowledge state (slip). The knowledge parameters, prior and learn rate, are the probability that a student knows the skill before answering any questions and the probability that, if the student is in the negative knowledge state, she will transition to the positive state at any given opportunity.

3 Related Work

There has been a limited amount of prior work focusing on detecting the moment of learning. We were able to track one relevant publication by Baker and colleagues [1]. They investigated detection of moment of learning in student data by modifying BKT structure. Another relevant result was published by [5]. They looked at scoring student model fits on simulated data and found a disparity between rankings of two frequently used metrics: root mean squared error and area under ROC curve. In this work we would like to address the question of the quality of detecting the moment of learning and investigate the problem of choosing a goodness-of-fit metric for that purpose.

4 Data

Our simulation dataset consisted of 1,000 simulated students and 100 skills with 30 questions per skill. Every student answered all 30 questions for each of the 100 skills. In the BKT simulation model we included no dependencies between skills and also no student specific parameters; therefore, the data can be thought of as either being produced by 1,000 students total or a new 1,000 students for every skill. Programmatically, data for each skill is stored in a separate file. Each row in each file represents one students data for that skill. The data stored from the simulation contains the students ground truth binary state of knowledge (mastered or not) at each of the 30 opportunities to answer (first 30 columns) and also the students correctness of responses to the 30 questions (stored in the second set of 30 columns).

In addition to the simulated data files containing student knowledge states and observed responses, we had corresponding files containing inferences of knowledge states and predictions of responses made with 16 different parameter sets resulting in 1,600 prediction files. Details of the parameter selection for simulation and prediction are discussed in the next section.

5 Methodology

5.1 Simulation

We generated 1,000 students knowledge and performance for 100 skills. Skills are defined by a set of four knowledge tracing parameters which the skill data is generated from. The 100 sets of four parameters were selected at random, uniformly sampling from the following constrained ranges for the parameters; prior between 0.01-0.80, learn rate between 0.01-0.60, and guess and slip between 0.05-0.40. After the 100 sets of parameters were selected, simulated data was produced by specifying a Dynamic Bayesian Network representation of Knowledge Tracing with a time slice length of 30. This representation, defined in Kevin Murphys Bayes Net Toolbox, with a particular parameter set fixed in the conditional probability tables, was then sampled 1,000 times, representing each simulated student. The sample Dynamic Belief Network function in BNT for simulation is a simple one; a random number between 0 and 1 is generated, if the number is equal to or lower than the prior parameter, the simulated student begins in the negative (not learned) state at time slice 1. To generate the observed response at this time slice, another random number is generated, if that number is greater than the guess parameter, the observed response is incorrect. To determine if the students knowledge state is positive (learned) at the next time slice; a random number is generated, if that number is less than or equal to the learning rate, then the students state is positive. With a positive state, the new random number needs to be greater than the slip parameter in order to produce a correct response. This is repeated for 30 times to simulate 30 knowledge states and observed responses per student.

5.2 Prediction

Typically, to predict student data, a hold-out strategy is used whereby a fraction of the students and their data is used to find a good fitting set of parameters. That good fitting set is then used to predict the fraction of students not used in training. The research question of this paper did not involve parameter fitting but rather required us to evaluate various models and observe how the models prediction of performance corresponded to its inference of knowledge. To do this we needed variation in models which we accomplished by choosing 16 candidate parameter sets with which to predict student data from each of the 100 skills. Since no training was involved, all data served as the test set. The top five sets of parameters used in the Cognitive Tutors was used, as well as 10 randomly generated parameters sets using the the same parameter constraints as the simulation, and, lastly, the ground truth parameter set for the skill was used to predict. The the same 15 parameter sets were used to predict the 100 skills, only the ground truth parameter set changed.

The prediction procedure is the same one used in all papers that use Knowledge Tracing; the prior, guess and slip parameters dictate the probability of correct on the first question. After the prediction is made, the correctness of

Table 1: Confusion Table

		Actual	
		Correct	Incorrect
Predicted	Correct	True Topisitve(TP)	False Positive (FP)
	Incorrect	False Negative (FN)	True Negative (TN)

the first question is revealed to the KT algorithm, which incorporates this observation using Bayes Theorem to infer the likelihood that the knowledge was known at that time. A learning rate transition function is applied and the processes is repeated 30 times in total to create 30 predictions of knowledge and 30 predictions of correctness per student for a skill.

6 Metrics

The most common metrics used to evaluate prediction performance in the EDM literature has been Area Under the Receiver Operator Curve (AUC) and Root Mean Squared Error (RMSE). One of the goals of our experiment is to reveal how indicative these measures are of the models accuracy in inferring knowledge. While these are the most common metrics, many others have been used in machine learning to evaluate predictions. We utilize a suite of metrics to investigate which metric is best at forecasting knowledge inference accuracy.

6.1 Model Performance

We selected a set of metrics in wide use today to score models when predicting student performance and knowledge state. Below is a short description of them.

Confusion Table Metrics Confusion table (rf. Table 1) is a table widely used in information retrieval and is a basis for a set of metrics capturing correctness of a retrieval or classification algorithm. Rows and columns of the confusion table denote the predicted and actual classes respectively and the cells in the intersection contain the counts of cases. Refer to Table 1 for an illustration. Here we illustrate a case for binary classification akin to the problem of binary classification of student performance (correct or incorrect) and state of knowledge (known or unknown).

If prediction is not categorical, say a probability from [0, 1], it is customary to round it: probabilities of 0.5 and greater become 1. For example, the cases when prediction matches the reality are captured in True Positive cell and the cases when the actually incorrect responses are marked as correct are captured in False Positive cell. We will use the confusion table metrics below.

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1a}$$

$$precision = \frac{TP}{TP + FP} \tag{1b}$$

$$recall = \frac{TP}{TP + FN} \tag{1c}$$

$$F - measure = 2\frac{precision\dot{r}ecall}{precision + recall} \tag{1d}$$

As opposed to the so-called point measures described above, there is also a frequently used Area Under Receiver Operating Characteristic curve (AUROC), which is a curve measure. The curve is produced by varying the rounding threshold (0.5 for point measures) from 0 to 1 and computing and plotting False Positive Rate (FPR) vs. True Positive Rate (TPR) (see below).

$$TPR = \frac{TP}{TP + FN} \tag{2a}$$

$$FPR = \frac{FP}{FP + FN} \tag{2b}$$

An area under resulting curve is the sought metric. An area of 0.5 is equivalent to random chance for a binary classifier. An area greater than 0.5 is, thus, better than chance. An exact AUC calculation can also be derived by enumerating all possible pairs of predictions. The percentage of the pairs in which the true positive prediction is higher is the AUC. This is the ability of the predictor to discriminate between true and false.

Pseudo $\boldsymbol{R^2}$ R^2 or percent variance explained is often used as a goodness of fit metric in linear regression analysis. For with binary classification, there exist several versions of R^2 called pseudo R^2. Applicable to our situation is Efrons pseudo R2 (refer to Equation below).

$$R^2 = 1 - \frac{\sum_{i=1}^{N} y_i - \hat{y_i}}{\sum_{i=1}^{N} y_i - \bar{y}} \tag{3}$$

Where N is the number of data points, y_i is the i-th component of the observed variable, $\bar{y_i}$ is the mean observed value, and $\hat{y_i}$ the prediction of i-th component of the observed variable.

Metrics Based on Log-Likelihood Likelihood functions are widely used in machine learning and classification. Likelihood captures the probability of the observing data given parameters of the model. In binary classification a natural log transformation of the likelihood function is often used (see below). Here

N is the total number of datapoints, y_i is the i-th component of the dependent variable, $\hat{y_i}$ is the predicted value of the i-th component of the dependent variable.

$$loglikelihood = \sum_{i=1}^{N} y_i \ln(\hat{y_i}) + (1 - y_i) \ln(1 - \hat{y_i}) \tag{4}$$

In addition to log-likelihood itself, there are several metrics that use log-likelihood as kernel component. For example, Akaike Information Criterion (AIC), Akaike Information Criterion with correction for finite sample size (AICc), Bayesian Information Criterion (BIC), and several others. These metrics introduce various forms of penalty for the size of the model (number of parameters) and number of datapoints in the sample in order to put overfitting models at disadvantage when performing model selection. Here k is the number of model parameters, N is the number of datapoints.

$$AIC = -2loglikelihood + 2k \tag{5a}$$

$$AICc = AIC + \frac{2k(k+1)}{N-k-1} \tag{5b}$$

$$BIC = -2loglikelihood + k\ln(N) \tag{5c}$$

Since we are comparing models that are only different in the parameter values and are doing so on the same dataset, we will not see difference in ranks assigned by log-likelihood, AIC, AICc, and BIC metrics.

Capped Binomial Deviance In addition to log-likelihood and log-likelihood-based metrics, we include the Capped Binomial Deviance (CBD). Capped binomial deviance is a version of the log-likelihood where prediction values are mandated to be at least away from 0 and 1 values and uses a logarithm with base 10 instead of natural logarithm. The is usually set to a small value of 0.001.

6.2 Moment of Learning

To capture the quality of detecting the moment of learning we devised a metric based on mean absolute deviation (MAD). Namely, moment of learning MAD is the average absolute difference of number of skill application opportunities between the moment when the internal state of the generating skill model switched to learned state and the moment when the probability of the skill being in a learned state reaches 0.95 (a traditionally used threshold in the area of intelligent tutoring systems). A perfect model would have a moment of learning MAD of 0. The larger the moment of learning MAD is the worse the model prediction of model of learning is.

7 Experiments and Results

7.1 Experiment 1

Research question: Among accuracy metrics used for ranking various parameter sets (models), which ones correlate best with accuracy of moment of learning prediction?

7.2 Results

The Table 2 below contains the correlations of performance prediction value, knowledge prediction value for all metrics, and moment of learning mean absolute error. Since prediction of performance is most widely adopted as a standard approach and the fact that we are trying to contrast it to the moment of learning mean absolute error, we sorted the rows corresponding to various statistical metrics by the respective column. The first column lists the metric used to evaluate the goodness of performance and knowledge prediction. The second column is the correlation between knowledge and performance prediction using the particular metric on both (this is the column the table is sorted by). The third column is the correlation between the particular metric used to evaluate performance and Mean Absolute Deviation (MAD) of Moment of Learning prediction. This is the column which tells us if the metrics used to evaluate performance are correlated with error in mastery / Moment of Learning prediction. The fourth column gives correlations of Moment of Learning MAD and metric values for predicting internal knowledge state. This correlation captures agreement between identifying the moment student learned a skill (this happens once per student-skill tuple) and the correctness of identifying the skills knowledge state for the student across all skill attempts.

7.3 Experiment 2

Hypothetically, the ground truth parameter sets should be the best at both making predictions of performance and estimating knowledge. A good metric should favor the ground truth parameters, therefore we ask: How often is the ground truth model the best at prediction performance according to the various metrics?

7.4 Results

The correlations of the performance and knowledge state prediction metrics from prior section targeted the 15 model parameter combinations that were different from the generating ground truth model parameters. Now, let us look at how the ground truth model compares to the other 15 we tested with respect to the statistical metrics we chose. Table 3, for each metric, gives the number of times a ground truth model parameter set is the best with respect to a given metric, and an average rank of the ground model parameter set as compared to the

Table 2: Metric correlations

Metric	Correlation of performance and knowledge metric	Correlation of performance metric and Moment of Learning MAD	Correlation of knowledge metric and Moment of Learning MAD
recall	0.878 ***	-0.954 ***	-0.819 ***
F-measure	0.561 ***	-0.839 ***	-0.792 ***
accuracy	0.522 ***	-0.802 ***	-0.822 ***
precision	0.334 ***	-0.797 ***	-0.628 ***
RMSE	0.470 ***	0.754 ***	0.828 ***
AIC	0.375 ***	0.751 ***	0.702 ***
AICc	0.375 ***	0.751 ***	0.702 ***
BIC	0.375 ***	0.751 ***	0.702 ***
CBD	0.409 ***	0.751 ***	0.762 ***
log-likelihood	0.375 ***	0.751 ***	0.702 ***
pseudo R^2	0.592 ***	-0.236 *	-0.296 **
AU ROC	0.335 ***	-0.119	-0.652 ***

Note: with respect to correlations with moment of learning MAD, in some cases a negative correlation is desirable (e.g., for accuracy), and for some cases a positive correlation is desirable (e.g., for RMSE). This is due to the fact that the smaller the moment of learning MAD the better, which is true for some metrics and the inverse is true for others. The table is sorted while observing this phenomenon (effectively sorting by the absolute value of the correlation coefficient).

Table 3: ground truth model rank vs. the other 15 models

Metric	Ground truth model has rank of 1	Mean rank of ground truth model
AIC	88/100	1.82/16
AICc	88/100	1.82/16
BIC	88/100	1.82/16
CBD	88/100	1.82/16
log-likelihood	88/100	1.82/16
RMSE	88/100	1.82/16
pseudo R^2	88/100	1.83/16
accuracy	33/100	2.52/16
F-measure	12/100	4.27/16
AU ROC	26/100	4.35/16
recall	0/100	6.65/16
precision	5/100	9.71/16

other 15 model. In each case we are aggregating across 100 different sets of 15 models plus one ground truth model. As we can see log-likelihood based models and RMSE form a group of metrics that gives ground truth models a large edge over the 15 reference models. Confusion table metrics, Area under ROC curve and the pseudo R2 gibe a drastically smaller support for it.

7.5 Experiment 3

Ground truth parameters do not always predict the data the best, but often do when using metrics like RMSE or log-likelihood. Do the parameter sets that are not predicted well by ground truth share a common pattern? Does the relative performance of ground truth correlate with high or low values of prior, learn, guess or slip in the generating parameters?

7.6 Results

Seeing log-likelihood based and RMSE metrics score the ground truth model at the same level of mean rank, we are wondering whether, across all 100 of generating parameter sets, the data produced by the same sets of parameters is equally hard to predict with ground truth model. For that we looked at whether the BKT parameter values correlate with ranks ground truth model receives on the moment of learning MAD metric.

First of all, moment of learning MAD metric ranked ground truth as best only 33/100 times with an average rank of 2.53/16. Correlations of moment of learning MAD ranks for ground truth models showed that theres a small marginally significant effect of pInit probability on the moment of learning MAD score ($r = 0.18$, $p - val = 0.07$). Guessing probability does not correlates with moment of learning MAD ($r = -.06$, $p - val = 0.55$).

Probability of learning and slip probability, however, are very strongly related to the moment of learning metric. The larger the learning rate of a simulated skill is, the higher the rank of the ground truth model is ($r = 0.68$, $p - val < 0.001$). Namely, the faster the skill is learned, the worse job ground truth model is doing. In the case of pSlip, the relation is the opposite: the higher the guess rate is, the higher rank moment of learning MAD assigns to the ground truth model ($r = -0.52$, $p - val < 0.001$).

Both the pLearn and pSlip parameters are controlling the process of skills transitioning into the learned state. Strong negative correlation of moment of learning MAD and pSlip is quite logical. Higher pSlip results in more errors even when the skill is mastered, as a result the transition to the learned state becomes more blurred. In this situation the ground truth model has an edge over other models. However, it is high to explain a high positive correlation of moment of learning MAD and pLearn. Higher pLearn means more correct responses overall, this should put ground truth model at an advantage. Additional investigation is necessary to address this phenomenon.

8 Discussion

In our first experiment we found that three less commonly used accuracy metrics showed the best correspondence to accuracy of moment of learning estimation. These metrics were: recall, F-measure, and accuracy, with recall giving a very high correlation of 0.954. Also noteworthy was the poor performance of AUC with a correlation of -0.119. This was the worst correlation and suggests that AUC should not be used to determine the relative goodness of models based on prediction performance if the underlying goal is to rank models based on knowledge estimation goodness. Metrics like recall and F-measure ought to be adopted in place of AUC for these purposes.

We also found that ground truth model parameters did not always perform the best and that RMSE and log-likelihood based metrics tended to predicted ground truth being the best parameter set more than the others. AUC, recall, F-measure, and precision, however, were among the worst. Therefore, if the underlying goal of an analysis is to recover ground truth parameters (such as with inferring pedagogical efficacy), RMSE and log-likelihood measures should be used and the aforementioned accuracy metrics should be avoided. The experiments 2 raised the question of why ground truth may not always predict the best experiment 3 indicated that high learning rate and low slip in the generating parameters can prove difficult for mastery prediction.

Overall detecting the moment of learning in the generated data by observing a switch from a string of all 0s (unknown state) to the string of all 1s (known state) is often not easy even when ground truth parameters are used. Especially if guess and slip parameters are larger, several back-and-forths between known and unknown state are common. In the area of ITS it is customary to wait till three correct attempts in a row to be sure student has mastered the underlying skill. In our case, when we assumed the moment of learning is the first time when probability of knowing the skill crosses the 0.95 threshold. Following from recent results on the lag with detecting the moment of learning that occurs in the Bayesian Knowledge Tracing [3], in future, we will experiment with adjustments to our computation of the moment of learning to compensate for this.

References

1. Baker, R.S.J.d., Goldstein, A.B., Heffernan, N.T. (2010) Detecting the Moment of Learning. Proceedings of the 10th Annual Conference on Intelligent Tutoring Systems, 25-34.
2. Corbett, A. T. and Anderson, J. R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modeling and User-Adapted Interaction, 4(4), 253-278. (1995)
3. Fancsali, S.E., Nixon, T., Ritter, S. (2013) Optimal and Worst-Case Performance of Mastery Learning Assessment with Bayesian Knowledge Tracing. In: Proceedings of the 6th International Conference on Educational Data Mining.
4. Koedinger, K. R., Anderson, J. R., Hadley, W. H., and Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. International Journal of Artificial Intelligence in Education, 8, 3043.
5. Pardos, Z. A., Wang, Q. Y., Trivedi, S. (2012) The real world significance of performance prediction. In Proceedings of the 5th International Conference on Educational Data Mining. Crete, Greece. pp 192-195.

Impact of Prior Knowledge and Teaching Strategies on Learning by Teaching

Ma. Mercedes T. RODRIGO[1], Aaron ONG[1], Rex BRINGULA[2], Roselle S. BASA[2], Cecilo DELA CRUZ[2], Noboru MATSUDA[3]

[1]Ateneo Laboratory for the Learning Sciences, Department of Information Systems and Computer Science, Ateneo de Manila University, Loyola Heights, Quezon City, Philippines
[2]University of the East, Manila, Philippines
[3]Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA, USA
mrodrigo@ateneo.edu, icemanfresh@yahoo.com
rexbringula@gmail.com, roselle_basa@yahoo.com
noboru.matsuda@cs.cmu.edu

Abstract. We investigate cognitive factors that are predictive of learning gains when students learn to solve equations by teaching a synthetic peer, called SimStudent. Previous empirical studies showed that prior knowledge is strongly predictive of post-test scores. However, in a recent study in the Philippines that replicated our previous study in the USA, there were students with low prior-knowledge who tutored their SimStudent better than other equally low prior students. In this paper, we analyze both process data (tutoring interactions) and outcome data (test scores) to understand what makes learning by teaching more effective. The results imply a presence of individual behavioral differences beyond the difference in the prior knowledge that might have affected SimStudent's learning, which in turn had non-trivial influence on tutor learning.

Keywords. Learning by teaching, teachable agent, SimStudent, Algebra equations, prior knowledge

1. Introduction

Since the late 1990s, researchers have investigated intelligent tutoring systems with intelligent pedagogical agents (often called *teachable agents*) to study a promising type of learning where students learn by teaching [1-3]. These technologies allow researchers to conduct tightly controlled experiments and to collect detailed *process data* representing interactions between students and teachable agents that together provide empirical evidence for the benefit of learning by teaching [4].

Matsuda *et al.* (in print), for example, showed that students' learning significantly correlated with the learning of teachable agents. Biswas *et al.* [5]

studied whether students could learn to self-regulate their teaching activities and how the ability of self-regulation affects the tutor learning. It is therefore of intellectual interest to uncover how the tutoring interaction affects students' learning by teaching.

In the current study, we use SimStudent, which is a teachable agent that helps students learn problem-solving skills by teaching [6]. It has been tested and redesigned several times, resulting in insights regarding the effects of learning by teaching and related cognitive theories to explain when and how students learn by teaching. Previous studies showed that pre-test score were highly predictive of post-test scores when students learn equation solving by teaching SimStudent [7]. In general, when students do not have sufficient prior knowledge on the subject to teach, they are not able to teach correctly and appropriately hence the benefit of learning by teaching would be arguably decreased.

Nonetheless, there are some students with low prior knowledge who learned more than others by teaching SimStudent. Among equally low-prior students, those who showed better performance on the post-test actually tutored their SimStudent better as well. The difference in the learning gain among students with comparable prior-knowledge indicates a presence of effective interaction for learning by teaching that might bootstrap tutor learning even with insufficient prior knowledge.

The goal of this paper is to investigate cognitive factors that affect tutor learning. The central research question is why some students (even with low prior knowledge) learned more than other students with comparable prior knowledge. To address this research question, the current paper analyzes data from two classroom (in-vivo) studies conducted in the USA and the Philippines. The Philippines study was a replication of the USA study reported earlier [8].

In the rest of the paper, we first introduce a learning environment in which students learn to solve linear equations by teaching SimStudent. We will then introduce two classroom studies conducted in the USA and the Philippines followed by the results and discussions.

2. Online Learning Environment with SimStudent

This section provides a brief overview of SimStudent and the online learning environment, Artificial Peer Learning environment using SimStudent (APLUS), in which students learn to solve algebra equations by interactively teach SimStudent. Technical details about SimStudent and APLUS can be found elsewhere [7]

2.1. SimStudent

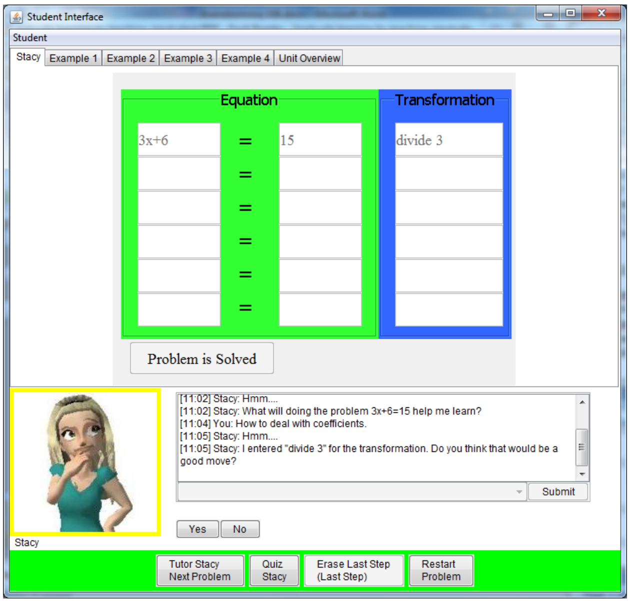
SimStudent is a synthetic pedagogical agent that acts as a peer learner. It learns procedural skills from examples. That is, a student gives SimStudent a problem to solve. SimStudent then attempts to solve the problem one step at a time, occasionally asking the student about the correctness of each step. If SimStudent cannot perform a step correctly, it asks the student for a hint. To respond to this request, the student has to demonstrate the step.

Students may not be able to provide the correct feedback and hints. As SimStudent is unable to distinguish correct from incorrect feedback, it continues to try to generalize examples and generate production rules that represent the skills learned. SimStudent is also capable of making incorrect inductions that would allow SimStudent to learn incorrect productions *even when students teach SimStudent correctly*. SimStudent's ability to model students' incorrect learning is one of the unique characteristics of SimStudent as a teachable agent.

2.2. APLUS: Artificial Peer Learning Environment using SimStudent

Figure 1 shows an example screen shot of APLUS. In APLUS, students act as a tutor to teach SimStudent how to solve equations. SimStudent is named Stacy and visualized at the lower left corner of APLUS. The tutoring interface allows the student and Stacy to solve problems collaboratively. In the figure, a student poses the problem 3x+6=15 for Stacy to solve. Stacy enters "divide 3" and asks the student whether this is correct. The student responds by clicking on the [Yes/No] button. If the student gets stuck, she can consult the examples tabbed at the top of the screen.

The student has the option of gauging how much Stacy has learned with the use of a quiz. The student chooses when and how often to administer



3x+6 = 15 divide 3

[11:02] Stacy: Hmm....
[11:02] Stacy: What will doing the problem 3x+6=15 help me learn?
[11:04] You: How to deal with coefficients.
[11:05] Stacy: Hmm....
[11:05] Stacy: I entered "divide 3" for the transformation. Do you think that would be a good move?

Fig 1. A screen shot of APLUS. SimStudent is visualzed with an avatar image and names Stacy.

the quiz by clicking a button at the bottom of the interface. The quiz interface looks like the tutoring interface, however, when Stacy takes the quiz, she does so independently, without any feedback or intervention from the student. At the end of the quiz, the student is presented with a quiz result.

The quiz is divided into 4 sections, each with two equation problems. The quiz items were created from the mix of one-step, two-step, and target equations (i.e., the equations with variables on both sides).

Stacy cannot progress to a section until she passes the previous section. The students were asked to tutor Stacy to be able to solve equations with variables on both sides. In the classroom studies, the students were informed that their goal was to help Stacy pass all four (4) sections of the quiz.

3. Methods

3.1. Participants

The USA study took place in one high school in Pittsburgh, PA, under the supervision of the Pittsburgh Science of Learning Center [8]. There were eight Algebra I classes with an average of 20 students per class. A total of 160 students with ages ranging from 14 to 15 participated in the study.

The Philippines study took place in one high school in Manila, Philippines, under the supervision of the co-authors from the University of the East and the Ateneo de Manila University. We enlisted participation from five first year high school sections with an average of 40 students per class. There were 201 study participants in all with ages ranging from 11 to 15. The average age of the participants was 12.5 years.

3.2. Structure of the study

In both the USA and the Philippine studies, each participant was randomly assigned to one of two versions of SimStudent: an experimental condition in which Stacy prompted the participants to self-explain their tutoring decisions and a control condition with no self-explanation prompts. The study was designed this way to investigate a particular research question on the effect of self-explanation for tutor learning [8], which is beyond the scope of the current paper. For three consecutive days, participants used their assigned version of SimStudent for one classroom period per day (42 minutes for the USA and 60 minutes for the Philippines study).

3.3. Measures

Students took pre- and post-test before and after the intervention. The students also took a delayed-test two weeks after the post-test was administered. Three versions of isomorphic tests were randomly used for pre-, post-, and delayed-tests to counterbalance the test differences. Students had the entire class period to finish the tests.

The tests are divided into five parts. Of these five parts, three parts are to test procedural knowledge on how to solve equations (the Procedural Skill Test, or PST), whereas other two parts are to test conceptual knowledge about algebra equations (the Conceptual Knowledge Test, or CKT). 102 out of 160 USA participants took all three tests, whereas in the Philippines 146 out of 201 participants took all three tests. In the following analyses, unless otherwise indicated, only those students who took all three tests are included.

The system automatically logged all of the participants' activities including problems tutored, feedback provided, steps performed, examples reviewed, hints requested, and quiz attempts. In the following analysis, we use these factors as process data.

4. Results

4.1. Overall Test Scores

Table 1 shows mean test scores plus or minus SD for the pre, post, and delayed Procedural Skill Tests from two studies. To see how students' test scores varied before and after teaching SimStudent, we conducted a two-way repeated-measures ANOVA with condition as a between-subjects variable and test-time (pre, post, and delayed) as a within-subjects variable. For the USA study, the repeated measure analysis revealed a weak trend for the main effect for test-time. A post-hoc analysis detected a difference from pre-test to post-test [8]. In the Philippines study, the test-time was also the main effect, and the post-hoc analysis detected that delayed-test was significantly higher than pre-test; $t(247.1) = 2.457$, $p < 0.05$. This difference, however, was likely due to the classroom instruction that students were taking during the two-week interval between the intervention and the delayed test.

Both in the USA and the Philippine studies, condition was not the main effect—the presence of self-explanation did not affect tutor learning with the version of APLUS and SimStudent used in two studies.

Table 1: Mean test scores **±** SD for pre, post, delayed procedural skill test for each study.

	Pre-test	Post-test	Delayed-test
Philippines (PH)	0.21±0.01	0.22±0.02	0.25±0.03
USA (US)	0.68±0.04	0.71±0.05	0.69±0.06

4.2. Impact of prior knowledge

As shown in Table 1, there was a notable difference in the pre-test scores suggesting that USA students had higher level prior knowledge than Philippine students; $t(142.4) = -22.25$, $p < 0.001$.

To see how prior knowledge affected learning and if the impact of prior knowledge differ between two studies, we ran a regression analysis with post-test score as a dependent variable and study (US vs. PH) as a fixed factor using pre-test score as a covariate. The results showed that pre-test is a strong predictor of post-test; *t*(244) = 2.80, *p* < 0.01. There was also a strong interaction between pre-test and study; the regression coefficient (slope) differed significantly between two studies; $b_{PH} = 0.32$ vs. $b_{US} = 0.76$; F(1,244) = 11.24, p < 0.001—suggesting that, in general, USA students gained (from pre- to post-test) more than Philippine students. Figure 2 shows the scatter plot for pre-test (x-axis) and post-test (y-axis) scores. USA students (red triangles) had steeper regression line than Philippine students.

4.3. Quiz Results

In the USA study, 36 out of 102(35%) students made their SimStudents pass all four quiz sections. In the Philippines study, no students passed all four sections. At the best, only 7 out of 146 (5%) of Philippine students had their SimStudents pass quiz section 2.

In the Philippines study, there were 73 students who solved quiz item #1 correctly. Of those 73 students, 68 students solved quiz item #2 correctly (hence by definition passing quiz section 1). Of those 68, only 11 students passed quiz section 2 (i.e., solving the first four quiz items correctly).

One possible explanation for the Philippine students' poor performance on the quiz is that Philippine students have insufficient prior knowledge, as indicated by the low pre-test scores and the weak regression slope. A number of factors may account for the difference prior knowledge, including curricular and age differences.

Still, some Philippine students managed to solve the first four quiz items (i.e., passing the quiz section 2), while others did not. Why might this be so? The next section addresses this issue.

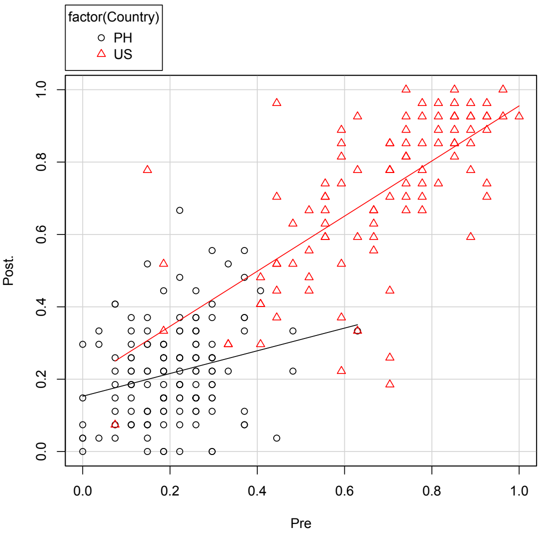


Fig. 2: Scatter plot of pre-test (x-axis) and post-test (y-axis) scores. US students had larger regression slope (0.76) than the PH students (0.32).

4.4. What makes learning by teaching more effective?

To understand why some SimStudents performed better on the quiz than others, we have analyzed the process data. In this analysis, we grouped students depending on the quiz sections their SimStudents passed. We call students whose SimStudents passed and failed quiz section *x* the "passing S*x*" and "failing S*x*" students, respectively. By definition, there were no passing S3 students in the Philippines study.

Our focus in this particular analysis is to understand how some students managed to pass quiz sections in the Philippines study. Therefore, we only included Philippine students for this analysis unless otherwise noted.

4.4.1. Accuracy of tutoring

One cognitive factor that had a significant contribution to tutor learning in the past studies is the accuracy of tutoring—i.e., the accuracy of recognizing correct and incorrect steps made by SimStudent as well as the accuracy the steps demonstrated as hint.

We thus compared the mean accuracy of passing/failing S1 and S2 students. The result suggested that the accuracy of tutoring is a key for success on the quiz in the Philippines study as well. For S1: $M_{Passing}$ = .70 (SD = .14) vs. $M_{Failing}$ = .52 (SD = 0.16); *t*(119.3)=-6.89, p < 0.001. For S2: $M_{Passing}$ = .75 (SD = 0.09) vs. $M_{Failing}$ = .59 (SD = 0.18); *t*(8.7)=-4.39, p < 0.01.

Students' prior knowledge should have affected tutoring accuracy. There was actually a strong correlation between the prior knowledge (measured as the pre-test score on the Procedural Skill Test) and the accuracy of tutoring. There was also a study difference—USA students tutored more accurately than Philippine students. The centered polynomial regression with the centered pre-test score (i.e., the difference from the mean) as the covariate (C.Pre) and the study (US vs. PH) as a fixed factor predicting the accuracy of tutoring (AT) revealed the following regression coefficients: AT = 0.62 + 0.16*C.Pre + 0.18[if US]; r^2=0.42, F(2, 235)=88.31, p<0.001; meaning that *Philippine students at the average procedural skill pre-test tutored with a 62% accuracy rate*. *USA students tutored 18% more accurately than Philippine students in general*. There was no study difference for the regression slope—suggesting that the prior knowledge affected the accuracy of tutoring equally in two studies.

A further analysis that compared passing and failing S1 students revealed that the prior knowledge was not the dominant factor that affected the accuracy of tutoring. In the Philippines study, the average pre-test score of the Procedural Skill Test for passing S1 students (*M*=.21, *SD*=0.10) was not higher than failing S1 students (*M*=.20, *SD*=0.09). However, the average accuracy of tutoring was higher for passing S1 students (*M*=.70, *SD*=.14) than failing S1 students (*M*=.52, SD=0.17).

As for the students' learning, there was a weak trend on the average normalized gain from pre- to post- favorable to passing S1 students (*M*=.05, *SD*=0.22) than failing S1 students (*M*=.01, *SD*=0.18); *t*(92.3)=-0.46, *p*=0.65.

This indicates that *the passing S1 students in the Philippines study learned more by teaching than the failing S1 students although where was no significant difference of the prior knowledge among them*. There might have been difference in the way passing and failing S1 students tutored SimStudent. The next section shows the results on analyzing process data.

4.4.2. Tutoring strategies

Since quiz items were fixed, using quiz items for tutoring could be a good strategy to help SimStudent pass the quiz. Actually, in the USA study, passing S4 students showed a higher percentage of using quiz problems for tutoring (M_{US} = .95, SD = .11) than failing S4 students (M_{PH} = .59, SD = .42); $t(28) = -4.08$, $p < 0.001$.

Thus, we first investigated whether passing S1 and S2 students in the Philippines study used more quiz items for tutoring than failing S2 students. We found that only 47% (1826 out of 3898) problems tutored in the Philippines study were the quiz items. Philippine students did not copy quiz items for tutoring as often as the successful (i.e., passing S4) USA students.

If time on task were a crucial factor for learning by teaching, then students who tutored on more problems should have learn more than those who tutored on fewer problems. To test this hypothesis, we first analyzed if passing S1 students simply tutored more problems than failing S1 students. The average number of problems tutored was 28.9±14.6 for passing S1 students and 20.9±12.2 for failing S1 students. The difference was not statistically significant. There was no notable difference in the number of problems tutored between passing and failing S1 students.

4.4.3. Resource usage

Did passing S1 students self-learn the materials by using resources more than failing S1 students? When counting the number of times students referred to worked-out examples, there was actually a notable difference. The passing S1 students referred to worked-out examples more than failing S1 students; $M_{Passing\ S1}$ (N=52) = 164±116 vs. $M_{Failing\ S1}$ (N=79) = 106±94; $t(93.19) = -3.00$, $p < 0.01$.

Furthermore, passing S1 students copied more example problems for tutoring than failing S1 students; $M_{Passing\ S1} = 2.2$ vs. $M_{Failing\ S1} = 1.4$; $t(111.16) = -3.62$, $p < 0.001$. Even when students did not actually understand how to solve equations, they could simply copy worked-out examples line by line to tutor SimStudent, which should have certainly affected SimStudent's ability to pass the quiz.

There was also a significant correlation between the number of example problems tutored and number of times example tab were clicked; $r^2=0.36$, $t(133)=8.67$, $p < 0.001$—suggesting that *Philippine students were actually switching between tutoring interface and example tabs frequently when they were copying example problems and their solutions for tutoring*.

4.4.4. Predictor of learning

Since there were several factors that contributed SimStudent's and students' learning found in the data, we conducted a regression analysis to see how certain factors contributed to the post-test score on the procedural skill test. The following variables were entered in the regression model: pre-test score on the Procedural Skill Test, total number of problems tutored, total number of quiz items tutored, total number of examples viewed, total number of example problems tutored, accuracy of tutoring, and study.

The result showed that pre-test score, accuracy of tutoring (AT), and study were significant predictors of post-test score (PTS) on the Procedural Skill Test. When pre-test score was centered (C.Pre), the following regression coefficients were revealed: PST = 0.21 + 0.61*C.Pre + 0.23*AT + 0.14[if US]; $r^2 = 0.77$, $F(3, 234)=267.7$, $p < 0.001$. Since pre-test and accuracy of tutoring are highly correlated, dropping accuracy of tutoring from the model also showed an equally good fit: PST = 0.34 + 0.63*C.Pre + 0.34[if US]; $r^2 = 0.76$, $F(2, 245) = 399.3$, $p < 0.001$.

5. Discussions and Concluding Remarks

We found that the prior knowledge had a strong influence on tutor learning—if students do not have sufficient prior knowledge for tutoring, they would not benefit from tutoring as much as students who have appropriate prior knowledge. The regression model mentioned in the results section shows that prior knowledge is the dominating predictor of post-test score for the Procedural Skill Test.

Nonetheless, in the Philippines study, students who managed to have their SimStudent pass the first quiz section (i.e., the first two quiz problems) outperformed those who failed to do so on the post-test of the Procedural Skill Test (albeit the small effect size) even when there was no pre-test difference between passing and failing students. Students who tutored SimStudent better learned more. The same correlation between SimStudent's and students' learning was observed in previous studies [7].

These results indicate that some students had actually learned how to tutor better SimStudent via the actual tutoring interaction. We found that, in the Philippines study, students who managed their SimStudent to pass the first two sections of the quiz copied worked-out examples more often than those who failed to pass the quiz. Furthermore, those passing students reviewed the worked-out examples more often than failing students. Further investigation would be necessary to understand how to better assist students with low prior knowledge to learn by teaching.

Learning by teaching is a promising type of learning especially when combined with an advanced agent technologies. Yet, there are many to understand when and how students learn by teaching and how to best facilitate their learning with various individual differences.

6. Acknowledgements

The authors thank the Ateneo Laboratory for the Learning Sciences, Marc Lester Armenta, Regina Ira Antonette M. Geli, Victoria Keiser, Gabriel Jose G. Vitug, and Evelyn Yarzebinski. We thank the Department of Science and Technology Philippine Council for Industry, Energy, and Emerging Technology Research and Development (PCIEERD) for the grant entitled, "Development of Affect-Sensitive Interfaces" and the Engineering Research and Development for Technology (ERDT) program for the grant entitled, "Development of an Educational Data Mining Workbench."

7. References

1. Chin, D., et al., *Preparing students for future learning with Teachable Agents*. Educational Technology Research and Development, 2010. **58**(6): p. 649-669.
2. Pareto, L., et al., *A Teachable-Agent Arithmetic Game's Effects on Mathematics Understanding, Attitude and Self-efficacy*, in *Proceedings of the International Conference on Artificial Intelligence in Education*, G. Biswas, et al., Editors. 2011, Springer: Heidelberg, Berlin. p. 247-255.
3. Uresti, J.A.R. and B. du Boulay, *Expertise, Motivation and Teaching in Learning Companion Systems*. International Journal of Artificial Intelligence in Education, 2004. **14**(2): p. 193-231.
4. Roscoe, R.D. and M.T.H. Chi, *Understanding tutor learning: Knowledge-building and knowledge-telling in peer tutors' explanations and questions*. Review of Educational Research, 2007. **77**(4): p. 534-574.
5. Biswas, G., et al., *Measuring Self-Regulated Learning Skills through Social Interactions in a teachable Agent Environment*. Research and Practice in Technology Enhanced Learning, 2010: p. 123-152.
6. Matsuda, N., et al., *Learning by Teaching SimStudent – An Initial Classroom Baseline Study comparing with Cognitive Tutor*, in *Proceedings of the International Conference on Artificial Intelligence in Education*, G. Biswas and S. Bull, Editors. 2011, Springer: Berlin, Heidelberg. p. 213-221.
7. Matsuda, N., et al., *Cognitive anatomy of tutor learning: Lessons learned with SimStudent*. Journal of Educational Psychology, in print.
8. Matsuda, N., et al., *Studying the Effect of Tutor Learning using a Teachable Agent that asks the Student Tutor for Explanations*, in *Proceedings of the International Conference on Digital Game and Intelligent Toy Enhanced Learning (DIGITEL 2012)*, M. Sugimoto, et al., Editors. 2012, IEEE Computer Society: Los Alamitos, CA. p. 25-32.

AIED 2013 Workshops Proceedings
Volume 5

The 4th International Workshop on Culturally-Aware Tutoring Systems (CATS 2013)

Workshop Co-Chairs:

Emmanuel G. Blanchard
Department of Architecture, Design, and Media Technology
Aalborg University at Copenhagen, Denmark

Isabela Gasparini
Department of Computer Science
University of Santa Catarina State, Brazil

http://cats-ws.org/

Preface

The 4th international workshop on Culturally Aware Tutoring Systems (CATS2013) is a follow-up to the three previously successful CATS workshop editions, organized in conjunction with ITS2008, AIED2009, and ITS2010. It discusses the place of culture in AIED research. Considering culture in this field is important because it is known to have a strong impact on many cognitive and affective processes including those related to learning. Furthermore, people with different cultural backgrounds develop alternative interpretations and strategies and do not similarly appraise their environment, which naturally reflects in their interactions with AIED systems.

For the 2013 edition of the CATS workshop, it was decided to put a particular emphasis on addressing the following topics: i) designing AIED systems to teach cultural knowledge and intercultural skills, ii) enculturating AIED systems (i.e., developing AIED mechanisms that incorporate cultural features), and iii) considering cultural biases/imbalances in the AIED research production, and ways to deal with them.

The scientific quality of CATS2013 was ensured by an interdisciplinary program committee of 21 members representing 11 different countries and 4 continents. A total of five papers were accepted for presentation, and the workshop also includes an interactive panel discussion whose topic is: "*AIED in non-western environments: Challenges and Opportunities*".

We are most grateful to the many individuals who have made this half-day workshop possible. We thank the Program Committees of the International Conference on Artificial Intelligence in Education, especially workshop chairs Erin Walker and Chee Kit Looi for their help in the planning of this workshop. We wholeheartedly thank the members of the CATS 2013 Program Committee for having dedicated time to evaluate workshop submissions within a limited time frame.

Welcome to the 4th International Workshop on Culturally-Aware Tutoring Systems.

July, 2013
Emmanuel G. Blanchard and Isabela Gasparini.

Program Committee

Co-Chair: Emmanuel G. Blanchard, *Aalborg University at Copenhagen, Denmark*
(Emmanuel.g.blanchard@gmail.com)
Co-Chair: Isabela Gasparini, *University of Santa Catarina State, Brazil*
(Isabela.gasparini@udesc.br)

Ryan S.J.D. Baker, *Teachers College, Columbia University, USA*
Benedict du Boulay, *University of Sussex, UK*
Jacqueline Bourdeau, *TELUQ, Canada*
Stefano A. Cerri, *University of Montpellier, France*
Vania Dimitrova, *University of Leeds, UK*
Birgit Endrass, *Augsburg University, Germany*
Geneviève Gauthier, *University of Alberta, Canada*
Monique Grandbastien, *University of Nancy, France*
Seiji Isotani, *University of Sao Paulo, Brazil*
W. Lewis Johnson, *Alelo Inc., USA*
Stan Karanasios, *University of Leeds, UK*
Paul Libbrecht, *Martin Luther University of Halle, Germany*
Samuel Mascarenhas, *University of Lisbon, Portugal*
Riichiro Mizoguchi, *Advanced Institute of Science and Technology, Japan*
Amy Ogan, *Carnegie Mellon University, USA*
Elaine Raybourn, *Sandia Laboratories, USA*
Matthias Rehm, *Aalborg University, Denmark*
Katharina Reinecke, *Harvard University, USA*
Ma Mercedes T. Rodrigo, *Ateneo de Manila University, The Philippines*
Silvia Schiaffino, *Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina*
Dhavalkumar Thakker, *University of Leeds, UK*

Table of Contents

A Conceptual Model of Intercultural Communication: Challenges, Development Method and Achievements

Emmanuel G. Blanchard[1], Stan Karanasios[2], Vania Dimitrova[3]

[1]Department of Architecture, Design, and Media Technology
Aalborg University at Copenhagen, Denmark
Emmanuel.g.blanchard@gmail.com
[2]AIMT Research Group, Business School,
University of Leeds, UK
s.karanasios@leeds.ac.uk
[3]School of Computing, Faculty of Engineering
University of Leeds, UK
v.g.dimitrova@leeds.ac.uk

Abstract. This paper argues that there is a need for integrating cultural considerations into AIED systems in order to enhance interactions between systems and learners. The development of a conceptual model of intercultural communication, the challenges encountered and the major achievements are described.

Keywords: Tutoring systems, intercultural communication, conceptual model.

1 Introduction

There is a large body of evidence which shows that the way people interpret and react to their environment significantly differs from one culture to another [1, 2]. When considering the wide range of human activities and situations influenced by culture, it is surprising to note that human-related technologies have only recently started to account for culture; and the domain of Artificial Intelligence in Education (AIED) is no exception. Indeed, AIED systems tend to focus on questions identified in Western contexts, resulting in design and solutions essentially inspired by Western authors, tested and validated essentially on Western samples [3].

This cultural imbalance in AIED research production, put together with well-documented cultural variations in situational understanding, interactions, and communication practices [1, 2, 4-12] bring forth the importance of considering cultural variations in AIED research. Specifically, we argue that two additional areas of research should become priorities for the AIED community:

- Investigating the applicative boundaries of previous AIED findings, assessing their universality or cultural specificity; and, possibly, initiating specific international collaborations and reflections on the most appropriate approaches to achieve these objectives.

- Developing innovative mechanisms to create more truly Culturally-Aware Tutoring Systems capable of manifesting cultural intelligence [4] in their inner mechanisms and interactions with learners.

Following initial developments (see [13] for an overview), one important emerging question facing the AIED community is *how* to *enhance interaction between AIED systems and learners by integrating cultural considerations?* By presenting a theory-grounded conceptual model of intercultural communication, with a particular focus on its nonverbal component, we contribute to this overarching research question and towards bridging the culture divide in the extant AIED research. Our model provides an ecology of notions as generic guidelines and structures that, we believe, can underpin AIED-related developments such as a) innovative designs for embodied pedagogical agents to allow learners to adopt a culturally-inspired and informed non-verbal communication style (see [14] for an example of enculturated agents), b) the development of automatic observation mechanisms to more appropriately interpret learners' body language, or c) the development of educational data mining techniques to analyze the resulting data. The research was undertaken as part of the ImREAL project developing a lightweight ontology to be used for semantic tagging of culturally rich social-web content [15].

This paper is organized as follows. Next, we introduce challenges in undertaking cultural research and strategies to mitigate the associated risks. We then describe the methodology followed to produce a conceptual model for cultural variations in interpersonal encounters. Following an iterative ontology development methodology, the conceptual model progressively evolved to include more and more 'heavyweight ontology'-inspired development practices. The resulting conceptual model is then presented and discussed. The paper closes with limitations and concluding remarks.

2 Challenges and Mitigation Strategies in Cultural Research

Many specific challenges are faced when addressing the cultural domain in a scientific manner. Firstly, while culture is a common topic of discussion in everyday life it remains ill-defined. And people, including scholars, not literate in this field often tend to adopt folk conceptualizations without even noticing it. The existence of a large body of cultural theories and frameworks tailored to specific tools and practices and focused on different aspects, also contributes to the difficulty in developing a clear and coherent scientific approach to this domain. The daily manipulation of cultural knowledge is also essentially implicit, i.e. most of the time people are unaware that they are culturally acting or interpreting and, when they are, it can be particularly difficult for them to thoroughly describe the situation with folk language only. It is thus extremely important for a project to have scientifically-acknowledged groundings that should, if possible, reflect different theoretical perspectives in order to obtain the broadest possible view about a cultural research question.

The research presented here has considered several cultural theories and frameworks. The essential ones are listed in Table 1. Some of them further propose strategies to address the risk of relying on cultural stereotypes, another central challenge in cultural research.

Secondly, and as previously mentioned, people are frequently unaware that they are culturally interpreting information and such an 'unconscious' bias does not spare well-informed cultural researchers. Besides adopting a very cautious way of thinking, a possible solution for (at least partially) limiting this effect is to enforce collaborations between people with very different profiles that are then able to nurture the reflection process with enculturated experiences. This eventually results in the identification of way more cultural specificities.

Table 1. References and brief descriptions for theoretical groundings of the current project.

Main references	Theory, framework, or study aspect
Memetic Theory [16]	A theory that suggests that cultural evolution shares similarities with genetic evolution. It is centered around the notion of 'meme' as basic cultural units i.e. the cultural counterpart to 'gene'.
Dual Inheritance Theory. See [17] for an overview.	A prominent contemporary approach to culture in evolutionary anthropology.
Sperber's Epidemiology of Representation [5]	Another influential theory in evolutionary anthropology that does not imply the notion of cultural replicators.
Distribution of cultural conceptualizations [18]	A psychology-based discussion on the notion of cultural conceptualizations, and on their distributions within cultural groups.
Culture and Cognition [6]	A psychology-based overview of cultural influences on cognitive processes
System of Values of Hofstede [7]. See [8] for a 25 year review of related studies.	Originally developed in the field of business/leadership, it remains the most commonly used framework in attempts to integrate cultural considerations in technology.
GLOBE system of values [9]	A system of values including both group and individual analyses. The main challenger of Hofstede's approach in business and leadership.
Schwartz Value Inventory [10]	Another system of values.
Cultural Intelligence [4]	Construct proposed in business/leadership to express, assess and improve behavioural, cognitive and affective intercultural skills
Cultural framework of Alwood [19]	A cultural framework that includes, but is not limited to, considerations for intercultural communication.
Framework for intercultural training of Bennett [20]	An approach for intercultural training that proposes a developmental model of intercultural sensitivity.
Research on specific cultural variations (e.g. [11])	Research on cultural variations related particularly to emotion, facial expressions, and nonverbal behaviour.
Cultural Framework of Hall [12]	A cultural framework that suggests that *space, context* and *time* are essential dimensions to understand how people behave, communicate and impact on their living environment.
Politeness Theory [21]	A theory that suggests that there are universalisms in ways of ensuring politeness in interpersonal communication.

The research team of the work presented is multicultural (Australian, French and Bulgarian nationals, with additional life experiences in the UK, Greece, Canada, Namibia, Japan, Denmark, and Germany) and benefits from discussions with collaborators from India and Germany. It also has a multidisciplinary expertise (computer science and social-science with advanced theoretical knowledge in educational and cognitive psychology, anthropology, and communication), and includes experts in both 'lightweight' and 'heavyweight' ontology engineering [22].

3 A Hybrid Development Method

This heterogeneous expertise in ontology engineering is actually an interesting illustration of the needs for a conceptual framework on intercultural communication. Cultures are not always country-related, and can emerge in any communities, including scientific ones. For example, it can be said that members of the AIED community do share a mutual culture. Yet within this community, there are conceptualizations mainly shared by psychologists that are not necessarily adopted by computer scientists, and conversely. Similarly, people working on lightweight and heavyweight ontologies aim at producing an artefact they all refer to as 'ontology'. Yet the meaning they give to this term drastically differs, which leads to strong variations in typical development procedures. According to prominent ontologists [22, 23], while lightweight ontologists follow operational approaches to find a solution to a problem known a priori, heavyweight ontologists follow approaches similar to philosophy in an attempt to capture the true essence of a domain before even considering issues they could address with the resulting conceptualization.

Since this project collaboration was initiated by lightweight ontologists, the team first adopted a lightweight ontology development approach. However, with internal assessments identifying more and more complex conceptual issues, heavyweight ontology practices were incorporated progressively. This resulted in a hybrid artefact that cannot be fully considered as an ontology since it lacks significant details, which is why we refer to it as a 'conceptual model'. It appears as more formal than average lightweight productions without fully matching heavyweight ontology requirements.

The complexity brought with the inclusion of more heavyweight practices also led to more strictly characterize the conceptualization focus. Rather than tackling intercultural communication at large, it appeared more realistic to first concentrate essentially on its nonverbal component. Yet, basic conceptual structures have been identified to support future work in addressing intercultural verbal communication (e.g. cultural scripts. See [24]). The next sections describe the steps followed.

Step 1. Adopting a glossary-centered approach. Developing a knowledge glossary (KG) (or glossary of terms) consisting in a list of widely accepted terminologies and their definitions along with supporting references is a common practice to provide theoretical grounding to lightweight ontologies [25]. This quickly appeared to be a problematic approach for modelling the intercultural communication domain because of its multidisciplinary nature. Several issues were observed such as 'cultural discipline' communities relying on constructs with no counterparts in other communities, or terms being used in several disciplines but with different meanings associated to them. Furthermore, a large number of term candidates were identified, which made the task of obtaining a coherent KG difficult because of cognitive overload aspects.

Step 2. Eliciting term interdependencies and providing a graphical representation. The first revision focused on structuring elicited terms rather than just listing labels and their definitions. Furthermore, this structuring was made graphical through the use of a concept map program, i.e. labels of selected notions were organized as a taxonomy-like tree while definitions and references for each of these labels (i.e. the KG) remained stored in a separate table. This provision of a graphical and structured overview of the KG facilitated the process, and further helped to reduce the list of term candidates by facilitating the identification of different terms labelling the same notion. Yet the structure remained was not optimal. More precisely, term categories were clearly emerging but no widely accepted labels existed for them.

Step 3. Enhancing the structure with the inclusion of abstract notions. The next methodological revision consisted in creating abstract categories to optimize the structure obtained in Step 2. Definitions for these categories had to be created since they did not exist in any specific cultural disciplines, but rather emerged from various perspectives analyzed altogether. None of these categories could thus be associated to an exact reference but rather to a body of supporting references. The resulting graphically-supported structure of labels and its associated KG then began to look satisfying. However, we wanted to expose our conceptualization to more cultural perspectives in order to better address threats of unconscious biases in cultural interpretations and the corollary risks of oversimplifying the problem.

Step 4. Iteratively validating and revising the model with competency questions (CQ). The use of CQs is an approach proposed to test that a model correctly covers its domain [25]. Briefly summarised, CQs are questions related to the domain such as "*are women and men normally allowed to make casual contact, e.g., shaking hands*?". CQs were collected from external experts and provided a vehicle to assess whether the model integrated appropriate notions to address them. We contacted people with expertise on culture-related topics (2 from the US, 2 from Germany, 1 from the Netherlands, 1 from Brazil, and 1 from the Philippines) and collected a total of 95 CQs, which were then used to assess the coverage of the nonverbal intercultural communication by our conceptual model. Due to space constraints, we cannot fully describe the systematic procedure followed. Each step was performed separately by two experts, followed by an in depth discussion to address identified limitations. Many CQs went beyond the nonverbal component of intercultural communication, with the resulting conceptual model being able to address them as well.

CQs were applied in an iterative manner. We divided them randomly into 3 sets of questions. The 1st set was used to analyze the model we had obtained after Step3, which led to significant updates. The new model was then tested with the 2nd set and a limited number of additional conceptual updates were adopted in a second revision. The 3rd set was eventually applied with no significant conceptual changes, which we interpreted as a sign that our model had achieved a proper level of stability and domain coverage. We argue that this approach is adequate when conceptualizing a cultural problem since it is not possible to find a source that concentrates the whole cultural wisdom and production of Mankind. In other words, there may always be a cultural group with specific and unforeseen interpretations for specific behavioral primitives. However, because of the stability we achieved, we hypothesize that future updates would remain light and expect that our model is dynamic enough to easily accommodate such limited evolutions.

This is indeed another important improvement resulting from CQ-based assessments. We identified that several notions in our model rely on complex combinations of contextual dimensions. Rather than attempting to list all possible combination instances (which we are confident is an impossible task), we have revised our model to include an easy mechanism for including new context 'descriptors' when needed.
This is one of the elements we discuss in the next section on the resulting production.

4 Resulting Conceptual Model

Figure 1 presents a simplified overview of the resulting conceptual model with the main concepts being introduced in the following lines.

Firstly, **culture** is seen as a cognitive phenomenon that emerges at group level [17] (see [3]). The main support for its exclusively cognitive nature is that cultures evolve through social learning processes [5, 17]. Cultural artifacts and behaviors are thus not directly transmitted. Rather, it is the way to design/construct/perform/etc. them that is socially shared (see the notion of cultural script below). Several cognitive constructs emerge in our conceptual model (see Table 3) with the most important ones for non-verbal intercultural communication being:

- **cultural norms** as "*a kind of grammar of social interactions. Like a grammar, a system of norms specifies what is acceptable and what is not in a society or group. And analogously to a grammar, it not the product of human design and planning*" [26];
- **cultural scripts** as prototypical procedures to be performed in a specific context and for a specific purpose. They are scripts as defined by [27]. The 'cultural script' concept was first introduced in linguistics [24] and social sciences [28] and is being expanded as part of the More Advanced Upper Ontology of Culture (MAUOC) project to address the non-universal nature of many cognitive scripts ([29]; see [30] for an outdated version; see also [31]);
- **stereotypes** as belief structures that influence the processing of information about stereotyped groups and their members [32]. They are *"sustained by selective perception and selective forgetting*" [33 p.196], and are *"socially-supported, continually revived and hammered in, by our media of mass communication*" [33 p.200].

As a follow up, it is important to clarify when intercultural communication practices, languages, and act are cultural and when they are not. This is achieved by assessing their innateness: if they are innate to human being (i.e. not acquired through social learning processes), then they are not cultural elements, which led us to identify **behavioral primitives** (gesture, posture, eye gaze, facial expression) as non-cultural because a new born baby could actually perform such things. However, what a baby cannot do is to perform these actions while associating a socially-learnt meaning to them. Such an association of behavioral primitives and socially learnt meanings are cultural and we refer to them as **Cultural Body Language Act** (CBLA see Table 2).

Another aspect of our conceptual model refers to the notion of **context**. Indeed several meanings can be associated to a behavioral primitive. Knowing which one applies in a specific situation depends on the ability to correctly identify contextual dimensions. Similarly, several cultural norms may be regulating nonverbal communication at a certain time, and are tightly depending on the context of occurrence. There are countless different contextual situations worldwide and it would be impossible to come to an exhaustive listing. We have thus defined **descriptors** as lightweight constructs to facilitate contextual descriptions (for a more heavyweight approach to context, see [30]). Descriptors are terms referring to qualities, properties, conditions, functions, or situations to characterize a contextual dimension. Several descriptors can be used to characterize a context. Example of descriptors can be 'politeness', 'gift', 'privacy', etc. virtually any terms that users may want to use as characterizations. Of course, a controlled vocabulary of descriptors would be better and, following CQs analyses, we already suggest several abstract descriptor categories (see Figure 1).

Finally, several additional notions specific to nonverbal intercultural communication have been defined in the KG with the main ones being listed in Table 2.

Table 2. Limited list of of definitions for nonverbal communication notions

Cultural elements	Basic cultural units of information. Initially popularized under the 'meme' terminology from *Memetic Theory* [16]. Alternatives less supportive of the genetic-to-culture analogy have also been proposed in modern evolutionary anthropology theories like the *Dual Inheritance Theory* [17] and the *Epidemiology of Representation* [5].
Cultural non-verbal communication	Communication system shared by a cultural group and acquired by its members through social learning processes (not innate [17]) which do not make use of oral language (e.g. [11]).
Cultural body language act (CBLA)	Behavioral primitives (gesture, posture, gaze or facial expression) or sequences of them associated with meanings, this association resulting from a sociocultural (not innate) learning process. Gestures associated with meanings. May be used to enrich, clarify or elaborate our descriptions [34, 35]. Postures associated with meanings. A form of kinetic behavior, revealing important information on nonverbal communication and emotions. Facial expressions associated with meanings. May be used to display affective states, which can repeat, augment, contradict, or be unrelated to verbal statements. Affect displays can be intentional or unintentional. Through facial expressions we can communicate our personality, open/close channels of communication, complement/qualify other nonverbal behavior, and communicate emotional states [2, 36].
CBLA – abstract	Definitions of these abstract body language constructs focus either on the effect to be achieved, the functional objective, or features specific to instances of these abstract categories (see definitions of regulators, illustrators, adaptors, and emblems below).
Regulators	Maintain and regulate the back and forth nature of speaking and listening between two or more interactants. They are gesture movements that attempt to regulate a conversation: to shut someone up, bring others in, encourage them to continue etc [37, 38].
Illustrators	Intimately linked to spoken discourse - actions accompanying speech such as finger pointing and raised eyebrows. They accompany and may amplify speech.[36, 38].
Adaptors	Generally unconscious behavioral adaptations in response to certain situations. Actions used to act on objects or self-manipulative actions such as lip biting [36, 37].
Emblems	Have a specific verbal translation known by most members of the communicating group. Usually the direct verbal translation consists of a word or two or phrase. Used often deliberately with the conscious intent to spread a message [34-36, 38, 39].
CBLA- concrete	Clear and precise usage of specific (sequences of) behavioral primitives to convey a meaning in more or less specific contexts (e.g. agreement with head nodding, greeting act with handshake).
Cultural body Language	A system of CBLAs internalized by members of a specific cultural group.

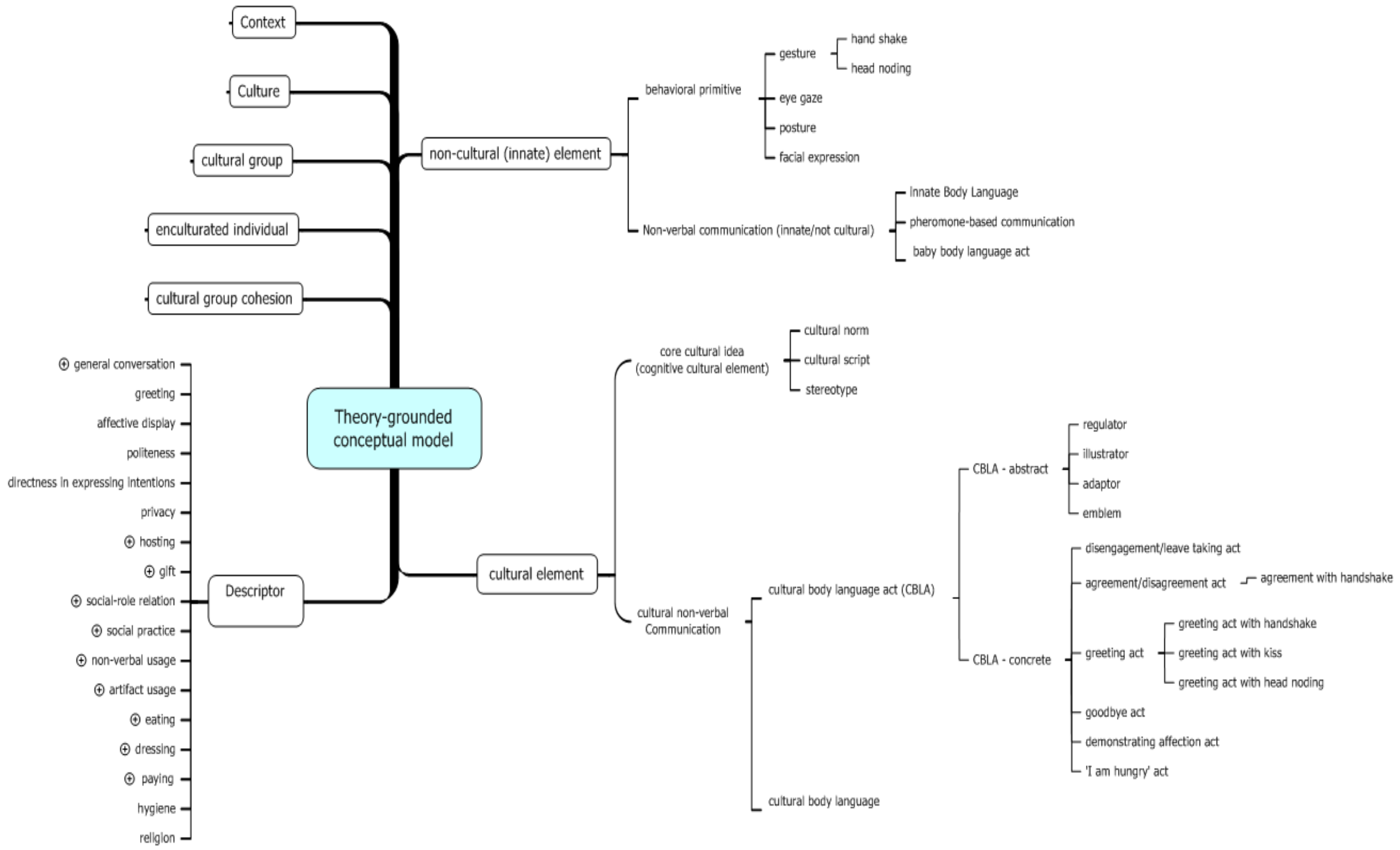


Fig. 1. Simplified graphical overview of the resulting conceptual model

5 Conclusion

We described the development of a conceptual model of intercultural communication in the context of addressing a cultural imbalance in the existent AIED research. The work presented is a step towards answering overarching research questions concerning how to enhance interactions between AIED systems and learners by integrating cultural considerations. As with all research that focuses on culture, some qualifications are in order. Whilst our research team encompasses a wide range of cultural backgrounds, we do not claim we account for every cultural perspective. The CQs captured the perspectives of 6 domain experts, producing 95 questions. Within the boundaries of our research we maintain that this was sufficient, however future research may build on this by including a broader perspective and greater volume. We have encoded the conceptual model in a lightweight ontology whose applicability for annotating user-generated content to capture cultural variations in nonverbal communication is currently evaluated. The conceptual model will also inspire heavyweight ontology developments in the context of the MAUOC project [29, 30].

6 Acknowledgement

This work received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no ICT 257831 (ImREAL project) and the Medialogy Department of Aalborg University.

References

[1] J. Henrich, S. J. Heine, and A. Norenzayan, "The weirdest people in the world?," *Behavioral and Brain Sciences,* vol. 33, pp. 61-83, 2010.

[2] B. Mesquita, N. H. Frijda, and K. R. Scherer, "Culture and emotion," in *Handbook of cross-cultural psychology:* . vol. 2, P. Dasen and T. S. Saraswathi, Eds., ed Boston: Allyn & Bacon, 1997.

[3] E. G. Blanchard, "On the WEIRD nature of ITS/AIED conferences: A 10 year longitudinal study analyzing potential cultural biases," in 11th International Conference on Intelligent Tutoring Systems (ITS2012), Chania, 2012. pp. 280-285.

[4] P. C. Earley and E. Mosakowski, "Cultural Intelligence," *Harvard Business Review,* vol. October, pp. 139-146, 2004.

[5] D. Sperber, *Explaining culture: A naturalistic approach*. Oxford: Blackwell, 1996.

[6] R. E. Nisbett and A. Norenzayan, "Culture and cognition," in *Stevens' Handbook of Experimental Psychology*. vol. 2, D. Medin and H. Pashler, Eds., 3rd ed New York: John Wiley & Sons, 2002.

[7] G. Hofstede, Hofstede G.J., and M. Minkov, *Cultures and organizations: Software in the minds*. New York: McGraw-Hill, 2010.

[8] B. L. Kirkman, K. B. Lowe, and C. B. Gibson, "A quarter century of culture's consequences: a review of empirical research incorporating Hofstede's cultural values framework.," *Journal of International Business Studies,* vol. 37, pp. 285-320, 2006.

[9] R. J. House, P. J. Hanges, M. Javidan, P. W. Dorfman, and V. Gupta, *Culture, Leadership, and Organizations: The GLOBE Study of 62 Societies* Thousand Oaks: Sage, 2004.

[10] S. H. Schwartz, "Beyond individualism/collectivism: New dimensions of values," in *Individualism and Collectivism: Theory Application and Methods*, U. Kim, H. C. Triandis, C. Kagitcibasi, S. C. Choi, and G. Yoon, Eds., ed Newbury Park, CA: Sage, 1994.

[11] D. Matsumoto and C. H. Hwang, "Cultural Similarities and Differences in Emblematic Gestures," *Journal of Nonverbal Behaviors,* vol. 37, pp. 1-27, 2013.

[12] E. T. Hall, *The Silent Language*. New York: Doubleday, 1983.

[13] E. G. Blanchard and A. Ogan, "Infusing Cultural Awareness into Intelligent Tutoring Systems for a Globalized World," in *Advances in Intelligent Tutoring Systems. Studies in Computer Sciences* R. Nkambou, R. Mizoguchi, and J. Bourdeau, Eds., ed Berlin: Springer, 2010, pp. 485-505.

[14] B. Endrass, M. Rehm, and E. André, "Towards culturally-aware virtual agent systems," in *Handbook of Research on Culturally-Aware Information Technology: Perspectives and Models*, E. G. Blanchard and D. Allard, Eds., ed Hershey, PA: IGI Global, 2010.

[15] S. Karanasios, D. Thakker, L. Lau, V. Dimitrova, D. K. Allen, and A. Norman, "Making sense of digital traces: from activity theory to ontology," *JASIST,* forthcoming.

[16] R. Dawkins, *The selfish gene*, 3rd ed. Oxford: Oxford University Press, 2006.

[17] J. Henrich and R. McElreath, "Dual inheritance theory: The evolution of human cultural capacities and cultural evolution," in *Oxford Handbook of Evolutionary Psychology*, J. Dunbar and L. Barrett, Eds., ed Oxford: Oxford University Press, 2007.

[18] F. Scharifian, "On cultural conceptualizations," *Journal of Cognition and Culture,* vol. 3, pp. 187-207, 2003.

[19] J. Allwood, "Intercultural Communication. English translation of: "Tvärkulturell kommunikation"," *Papers in Anthropological Linguistics 12,* vol. University of Göteborg, Dept of Linguistics, 1985.

[20] M. J. Bennett, "A developmental approach to training for intercultural sensitivity. Int. J. of Intercultural Relations," *Int. J. of Intercultural Relations,* vol. 10, pp. 179-196, 1986.

[21] P. Brown and S. C. Levinson, *Politeness: Some universals in language usage*. New York: Cambridge University Press, 1987.

[22] R. Mizoguchi, "Tutorial on ontological engineering - part 1: Introduction to ontological engineering.," *New Generation Computing,* vol. 21, pp. 365-384, 2003.

[23] B. Smith, "Ontology," in *Blackwell guide to the philosophy of computing and information*, L. Floridi, Ed., ed Oxford: Blackwell, 2003, pp. 155-166.

[24] C. Goddard and W. A., "Cultural scripts: What are they and what are they good for?," *Intercultural Pragmatics,* vol. 32, pp. 153-166, 2004.

[25] M. F. López, A. Gómez-Pérez, J. P. Sierra, and A. P. Sierra, "Building a Chemical Ontology Using Methontology and the Ontology Design Environment," *Ontologies,* pp. 37-46, 1999.

[26] C. Bicchieri and R. Muldoon. (2011, March 14). *Social Norms, Stanford Encyclopedia of Philosophy*. Available: http://plato.stanford.edu/entries/social-norms/

[27] R. C. Schank and R. Abelson, *Scripts, Plans, Goals, and Understanding*. Hillsdale, NJ: Earlbaum Assoc, 1977.

[28] H. C. Triandis, G. Marin, J. Lisansky, and H. Betancourt, "Simpatia as a cultural script of Hispanics," *Journal of Personality and Social Psychology,* vol. 47, pp. 1363-1375, 1984.

[29] E. G. Blanchard and R. Mizoguchi, "An introduction to MAUOC, the More Advanced Upper Ontology of Culture, and its interest for designing Culturally-Aware Tutoring Systems," *Research and Practice in Technology Enhanced Learning (RPTEL),* forthcoming.

[30] E. G. Blanchard , R. Mizoguchi, and S. P. Lajoie, "Structuring the cultural domain with an upper ontology of culture," in *Handbook of research on culturally-aware information technology: Perspectives and models*, E. G. Blanchard and D. Allard, Eds., ed Hershey PA: IGI Global, 2010.

[31] E. G. Blanchard, "Is it adequate to model the socio-cultural dimension of e-learners by informing a fixed set of personal criteria? ," in *12th IEEE Int. Conf. on Advanced Learning Technologies (ICALT2012)*, Roma, Italy, 2012, pp. 388-392.

[32] J. T. Jost and D. L. Hamilton, "Stereotypes in our culture," in *On the Nature of Prejudice: Fifty years after Allport*, J. Dovidio, P. Glick, and L. Rudman, Eds., ed Oxford: Blackwell, 2005, pp. 208-224.

[33] G. W. Aalport, *The nature of prejudice*. Cambridge, MA: Perseus Books, 1954/1979.

[34] A. Kendon, "Some topics in gesture studies," in *Fundamentals of verbal and nonverbal communication and the biometric issue*, A. Esposito, NATO Programme for Security through Science., and North Atlantic Treaty Organization. Public Diplomacy Division., Eds., ed Washington, DC: IOS Press, 2007, pp. 3-19.

[35] D. McNeill, *Gesture & Thought*. Chicago: University of Chicago Press, 2005.

[36] M. L. Knapp, *Nonverbal communication in human interaction*, 2nd ed. New York: Holt, Rinehart and Winston, 1978.

[37] P. Ekman and W. V. Friesen, "The repertoire of nonverbal behavior: Categories, origins, usage, and coding," *Semiotica,* vol. 1, pp. 49-98, 1969.

[38] A. Furnham, *Body Language at Work*. Channel Islands, UK: The Guernsey Press, 1999.

[39] T. Wharton, *Pragmatics and Non-Verbal Communication*. Cambridge: Cambridge University Press, 2009.

Is the Brazilian HCI community researching cultural issues? An analysis of 15 years of the Brazilian HCI conference

Isabela Gasparini [1,2], Marcos H. Kimura [1], Sergio L. de Moraes Junior [1],
Marcelo S. Pimenta [2], José Palazzo M. de Oliveira [2]

[1] Department of Computer Science,
Universidade do Estado de Santa Catarina (UDESC), Brazil
isabela.gasparini@udesc.br

[2] Institute of Informatics,
Universidade Federal do Rio Grande do Sul (UFRGS), Brazil
{mpimenta, palazzo}@inf.ufrgs.br

Abstract. This paper presents the results of a in-depth study to investigate if Brazilian HCI (Human Computer Interaction) community is addressing cultural issues. In this paper, the results emerge from a study of fifteen years paper production in Brazilian HCI conference. After this first analysis, this work explored each Brazilian' researcher curriculum from the previous result, aiming to understand how research field is evolving.

Keywords: HCI community, Brazilian conference, cultural issues.

1 Introduction

Brazil is a country with 8,514,876 km2, it is the largest country in both South America and the Latin America region and the world's fifth largest country, both by geographical area and by population, with over 193 million people. Brazil is 58th in the economic competitiveness ranking published by the World Economic Forum in September of 2010, which analyzed 139 countries [1]. Brazilian culture is a very extensive subject, including stories, legends, dances, superstitions and religious rituals, either brought to the land by the Europeans, Africans and Asians or already present in its native cultures. All of these manifestations are quite peculiar to each culture and distinct in each region of Brazil. Brazil is a multicultural and multiethnic society. Because of the Portuguese colonization, Portuguese is the official language and is spoken in the whole country; however, Brazilian culture has been influenced by many cultures, due to immigration throughout our history [2].

Human-computer interaction (HCI) is an area of research and practice that emerged in the early 1980s, as a specialty area in computer science embracing cognitive science and human factors engineering [3]. The initial research in the HCI field was motivated by the increase of personal computing that became manifest at an opportune time – personal computers were being used by end users who were not experts in computer science or engineering. As HCI developed, it moved beyond the desktop perspective. First, because of the growing influence of the Internet on computing and on society, and secondly, because HCI moved beyond the desktop through the continual, and occasionally explosive diversification in the ecology of computing devices. Nowadays, interactive systems can be anywhere and anytime. Therefore, today it is important to know how to deal with cultural issues, especially when developing or evaluating wide-access applications and interactive systems. Interactive systems for the Web need to provide support for an ever increasing amount of material and make it available for local-language populations across the world. One of the main challenges for designers is to build/evaluate system that aim explicitly at acknowledging the diversity of their users' cultural background and attending to a wider variety of needs and expectations [4]. Consequently, the introduction of the culture concept in interactive systems and interaction design is becoming a necessity, a challenge, and a timely and relevant issue [5]. Indeed, in attempting to disentangle this diversity, culture has received increasing attention in International Human Computer Interaction (HCI) community, *e.g.* in learning contexts [6] and [7]; in internationalization/globalization aspects [8], [9], [10] and [11]; in an adaptive user perspective [12]; in an usability evaluation [13]; in a Web Science Perspective [14]; in the software engineering [18]; and in HCI design [25]. For the last years, the research and literature accounting for cultural contexts in human-computer interaction design has quickly grown [25].

Nevertheless, until today, there is no analysis of how cultural issues have been addressed in Brazilian HCI community, neither how HCI community in Brazil has been working towards consolidating its cultural aspects in interactive systems. This paper presents an analysis of fifteen years of Brazilian Symposium on Human Factors in Computer Systems (IHC), to identify researchers who treat cultural issues as a core activity in their research. To do this, the conference between the years 1998 to 2012 was analyzed. After these results, we perform a qualitative analysis on each researcher who deals with cultural aspects to verify ongoing topic in Brazil.

This paper is structured as follows. Section 2 shows the related work. Section 3 describes our methodology of this work and Section 4 presents the results of cultural issues in Brazilian researches. Finally, section 5 presents the considerations of this analysis.

2 Related Work

One of the most accepted definition of culture is "the collective programming of the mind that distinguishes the members of one group or category of people from others" [26] and is usually defined in Human Computer Interaction as the common values, attitudes and behavioral patterns shared by a group of people [27]. Cultural awareness involves becoming aware of cultural values, beliefs and perceptions. It´s become central when we have to interact with people from other cultures. Blanchard et al [28] refer culturally-aware system to 'any system where culture-related information has had some impact on its design, runtime or internal processes, structures, and/or objectives'. The quality of user experience is intricately related to the users' cultural characteristics [15]. Cultural characteristics have been found to be an important issue because a user's cultural profile shapes his/her perception of a system features, e.g., a given culture profile will cause a user to focus on a set of information and ignore others, thus, system features appropriated for one culture may not be suitable for others; and system design needs to be adapted for different culture as well [15].

Considering culture in HCI interaction is becoming a necessity, a challenge, and a timely and relevant issue, as we can see the inclusion of this topic in different HCI conferences (*e.g.* the 14th International Conference on Human-Computer Interaction, that in 2011 had a tutorial about Cross-Cultural HCI [16], the 20th conference on User Modeling, Adaptation, and Personalization – UMAP that in 2009 selects as a best paper award a paper about adapting interfaces to Cultural Preferences [17], or the 14th IFIP TC13 Conference on Human-Computer Interaction - Interact 2013, that has the theme of the conference, "Designing for Diversity", which recognizes the interdisciplinary, multidisciplinary and intercultural spirit of human-computer interaction research). Cultural diversity is a hot topic to HCI today and many works have applied cultural dimension to interaction design, variations of dialog and presentation design, evaluation of user behavior, etc. [19], [20], [21], [22].

Regarding related works which aims to analyze data from conference papers, a paper stands out. Henry et al. [23] brings a general analysis of HCI conferences all over the world where he shows several graphic data, making possible a wide vision about how authors have behaved during years of work in that field of knowledge of computer science. Henry et al. [23] opens many possibilities of analysis which can show important information about how have being directed the researches all over the globe. They showed a visual exploration of the field of HCI through the author and article metadata of four of its major conferences: the ACM conferences on Computer-Human Interaction (CHI), User Interface Software and Technology, and Advanced Visual Interfaces and the IEEE Symposium on Information Visualization, and then they described many global and local patterns they had discovered in this data set, together with the exploration process that produced them.

Blanchard [6] analyzes potential cultural biases in paper production in Intelligent Tutoring Systems (ITS) and Artificial Intelligence in Education (AIED) conferences. The paper attempts to make the community aware of an identified and quantified WEIRD (*Western, Educated, Industrialized, Rich, and Democratic societies*) bias in psychology research that is likely to have an indirect impact on the AIED research field. A ten year analysis of full conference papers production reveals similar WEIRD imbalances in the AIED research field, which suggests that it may be producing WEIRD-flavored research as well [6].

Following the authors idea, this paper presents an in-depth study to investigate if Brazilian HCI community is addressing cultural issues in their researches. In this paper, the results emerge from *Brazilian Symposium on Human Factors in Computing Systems* (*IHC*) and the exploration of each Brazilian' researcher curriculum from the previous result, aiming to understand how research field is evolving. The next section will present the methodology of the analysis.

3 Methodology

This work focuses in the analysis of fifteen years of Brazilian Symposium on Human Factors in Computing Systems (IHC). To begin the review, all the data was reunited from all conferences editions and started a filtering of information. First decision was to investigate only full papers. Some editions (206-2012) were available in ACM Digital Library, but we also had access to the Proceedings of all conference editions, getting access to all papers. Then, we open each full paper, and put its information in a dataset (i.e., year, title, language, authors, institution, country, keywords, abstract, ACM keywords, ACM category, general terms, and references).

Also in parallel we inspected all 236 full papers, observing for each paper: the title, abstract, keywords and introduction, aiming identify the main subject of the work and if it treated cultural issues. Each paper identified as focusing with cultural issues, a fully reading was applied. Table 1 presents the number of papers investigated from each year. It is important to explain that the period from 2002 to 2010 the conference was biannual. In 2010 the community voted to return the annual conference. This year the Brazilian Symposium on Human Factors in Computing System will be placed at Manaus- the capital state of Amazonas – in the North of Brazil, because the community agrees that Brazil have diverse cultural components, and each region has its peculiarities, so the conference must be placed each year in a different region of Brazil. Since the others regions have already supported the conference (i.e. Northeast, Central-West, Southeast and South), the only region not yet covered was the North region.

Table 1. Number of full paper for each year of HCI conference

Conference year	*Place of the Conference (city and state in Brazil)*	*Number of full paper*
1998	Maringá – Paraná	15
1999	Campinas – São Paulo	13
2000	Gramado – Rio Grande do Sul	16
2001	Florianópolis – Santa Catarina	22
2002	Fortaleza – Ceará	29
2004	Curitiba – Paraná	15
2006	Natal – Rio Grande do Norte	20
2008	Porto Alegre – Rio Grande do Sul	25
2010	Belo Horizonte – Minas Gerais	19
2011	Porto de Galinhas – Pernambuco	32
2012	Cuiabá – Mato Grosso	30
Total		236

The articles were read individually, aiming searching for terms and themes that fits the specifications, *i.e.*, culturally related aspects. After that, we also did diverse queries in the dataset, aiming to confirm previous results with human inspection.

After this data stratification, we identified only six papers (related to sixteen researchers) of the conference treat directly cultural issues in their research. Therefore, we did an explorative search for the research of each researcher who published one of these papers. All these sixteen researchers who published cultural aspects in the conference work at Brazilian's institutes, so, as all of them must have an update curriculum vitae in the Lattes Platform provided by the National Council for Scientific and Technological Development (CNPq[1]) [24], the analysis of each curriculum was facilitated.

The understanding and analysis of this amount of data is not the core business, and this paper just initiates what seems to be an important issue of analysis at HCI field studies. Section 4 presents the results found.

4 Results

We analyzed 236 full papers of all Brazilian Symposium on Human Factors in Computing System editions. From this dataset, we had found a list of only six full papers that had focused on cultural related aspects as the core business of the paper. There were found other papers which have some relation to cultural aspects (e.g. accessibility, design for different types of users, etc.), but the cultural aspects were not treat or even cited in them. Table 2 presents the result of this analysis.

Table 2. Results of cultural aspects from the Brazilian Symposium on Human Factors in Computing System

Year of the conference	*Number of author*	*Cultural subject of the paper*
2000	3	Cultural and Psychological effects of Colors
2001	2	Methods for the study of signs perception for subjects in different cultural environment
2008	4	Cultural sensitive web-based learning material
2011	1	Cultural aspects to dealing with death issues and afterlife digital legacy
2011	3	Cultural-aware issues in HCI
2011	3	Cross-cultural systems and cultural Metaphors

From this small list of papers, we found sixteen researchers that deals with culture as a topic of research and investigation. After that we analyzed all Lattes curriculum vitae, in order to find out whenever these authors have also been publishing the cultural subject in other conferences or journals. The result of the triangulation analysis is shown in Figure 1.

[1] CNPq is an agency under the Ministry of Science and Technology (MST) which aims to promote scientific and technological research and also train and qualify researchers in the country and abroad

Fig. 1. Cloud tag of related conference where Brazilian researchers have been publishing the cultural subject.

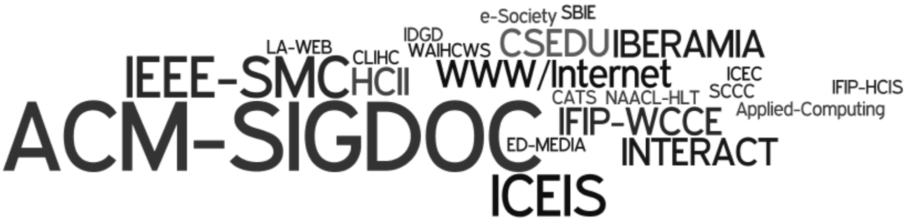


Table 3 presents the list of conferences where the researchers published papers with cultural related issues.

Table 3. Main conferences where Brazilian researchers published papers related to cultural issues

Acronym	*Name of the Conference*
ACM-SIGDOC	ACM International Conference on Design of Communication
Applied-Computing	IADIS International Conference Applied Computing
CATS	International Workshop on Culturally-Aware Tutoring Systems
CLIHC	Latin American Human- Computer Interaction
CSEDU	International Conference on Computer Supported Education
e-Society	IADIS International Conference e-Society
ED-MEDIA	World Conference on Educational Multimedia, Hypermedia and Telecommunications
HCII	International Conference on Human-Computing Interaction
IBERAMIA	Ibero-American Artificial Intelligence Conference
ICEC	International Conference on Entertainment Computing
ICEIS	International Conference on Enterprise Information Systems
IDGD	International conference on Internationalization, design and global development
IEEE-SMC	IEEE International Conference on Systems, Man, and Cybernetics
IFIP-HCIS	IFIP Human-Computer Interaction Symposium
IFIP-WCCE	IFIP World Conference on Computers in Education
INTERACT	IFIP TC13 Conference on Human-Computer Interaction
LA-WEB	Latin American Web Congress
NAACL-HLT	Young Investigators Workshop on Computational Approaches to Languages of the Americas
SBIE	Simpósio Brasileiro de Informática na Educação
SCCC	International Conference of the Chilean Computer Science Society
WAIHCWS	Workshop sobre Aspectos da Interação Humano-Computador na Web Social
WWW/Internet	IADIS International Conference WWW/Internet

In addition to the conference papers, some researchers have been publishing cultural aspects in different journals and books chapters. The most relevant are: International Reports on Socio-Informatics, Advances in Human-Computer Interaction and Human-Computer Interaction Series (Springer).

5 Conclusion

The Brazilian HCI community is very well consolidated in Brazil and is acknowledged both nationally and internationally. Members of the community feel that the people working in the field and their production have increased in the last few years [2]. This paper focused on carry out an exploratory search in the Brazilian Symposium on Human Factors in Computing System. From the two hundred thirty-six full papers study, only six were directly related to the cultural issues. After, we have summarized these findings and executed a new search toward to discover if the authors of these papers actually have been publishing about this subject in other conferences and journals. The result was positive, since the majority of them still working in the cultural aspects.

Our objective was not point out who in Brazil is researching about cultural issues and HCI, nor even show how much Brazilian HCI community is addressing cultural aspects (if this answer is possible or desirable) but to discuss how cultural issues can be addressed in HCI field. Indeed, the success of the growth of cultural issues in HCI research is exactly take advantage of workshops and conferences where the theme can be discussed.

HCI (as a community and as a research area) need more investigation towards this agreement. Many open issues emerge when we discuss culture in interaction design, and HCI, including:

- What exactly is culture? How we can represent it and use it appropriately in the interaction design?
- How do you obtain relevant cultural information about a specific community (country or region or even a corporation) and how do you determine each is relevant?
- How do you generate design ideas from this cultural information?
- How important is culture among all other aspects being considered in an interaction design?
- How Brazilian HCI community can address cultural issues in its research?

This paper obviously has not an answer to these questions, it just tries to provide some directions of how culture aspects still an open issue. Therefore, it is yet a limited excursion into a territory which includes many other possible perspectives and paths to explore.

6 References

1. Official Brazil Governmental website. Brazil in numbers. Available online at http://www.brasil.gov.br/sobre/brazil/brazil-in-numbers

2. Prates, R. O., Filgueiras, L. V. L. Usability in Brazil. I. Douglas and Z. Liu (Eds.) Global Usability. Human-Computer Interaction Series, Springer London, 2011. 91-109.
3. Carroll, John M. Human Computer Interaction - brief intro. In: Soegaard, Mads and Dam, Rikke Friis (Eds.). The Encyclopedia of Human-Computer Interaction, 2nd Ed. Aarhus, Denmark: The Interaction Design Foundation. 2013. Available online at http://www.interaction-design.org/encyclopedia/human_computer_interaction_hci.html
4. Salgado, L. C. de C., Sieckenius, C. de S., Leitão, C. F. On the epistemic nature of cultural viewpoint metaphors. In Proceedings of the 10th Brazilian Symposium on Human Factors in Computing Systems and the 5th Latin American Conference on Human-Computer Interaction (IHC+CLIHC '11). Brazilian Computer Society, Porto Alegre, Brazil, 2011, 23-32.
5. Gasparini, I., Pimenta, M.S., Palazzo, M., de Oliveira, J. Vive la différence!: a survey of culturally-aware issues in HCI. Proceedings of the 10th Brazilian Symposium on Human Factors in Computing Systems and the 5th Latin American Conference on Human-Computer Interaction (IHC+CLIHC '11). Brazilian Computer Society, Porto Alegre, Brazil, 2011, 13-22.
6. Blanchard, E.G. On the WEIRD nature of ITS/AIED conferences: A 10 year longitudinal study analyzing potential cultural biases. Proceedings of the 11th International Conference on Intelligent Tutoring Systems (ITS2012), Springer LNCS 7315, Chania, Greece, 2012, 280-285.
7. Limongelli, C., Sciarrone, F., Temperini, M., & Vaste, G. Virtual Cultural Tour Personalization by Means of an Adaptive E-Learning System: A Case Study. M.D. Lytras et al. (Eds.): WSKS2009, LNAI 5736, 2009, pp. 40–49, Springer-Verlag.
8. del Galdo, E., Nielsen, J. (Eds.). International User Interfaces. Wiley, New York, 1996.
9. Smith, A.; Yetim, F. (Eds.). Global Human–Computer Systems: Cultural Determinants of Usability. Interacting with Computers 16 (1) (special issue), 2004.
10. Amant, Kirk St. Linguistic and Cultural Online Communication Issues in the Global Age. IGI Global, 2007.
11. Moran, R. T., Harris, P. R., & Moran, S. Managing cultural differences - Global Leadership Strategies for the 21st Century. 7th ed., Elsevier, 2007.
12. Recabarren, M.; Nussbaum, M. Exploring the feasibility of web form adaptation to users' cultural dimension scores. User Model User-Adap Inter, 2010, 20, pp. 87-108.
13. Clemmensen, T., Hertzum, M., Hornbæk, K., Shi, Q., & Yammiyavar, P. Cultural Cognition in Usability Evaluation. Interacting with Computers, 2009, 21(3), pp. 212-220.
14. Lytras, M. D., Damiani, E., Carroll, J. M., Tennyson, R. D., Avison, D., Naeve, A., ... & Vossen, G. (Eds.) Visioning and Engineering the Knowledge Society - A Web Science Perspective. Proceedings of Second World Summit on the Knowledge Society, WSKS 2009, LNCS 5736, Springer, 2009.
15. Lee, I., Choi, G. W., Kim, J., Kim, S., Lee, K., Kim, D., ... & An, Y. Cultural Dimensions for User Experience: Cross-Country and Cross-Product Analysis of Users' Cultural Characteristics. Proceeding of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction, BCS-HCI '08, v. 1, 2008, pp. 3 -12
16. HCI2011 - 14th International Conference on Human-Computer Interaction, Tutorial Program - T08: Cross-Cultural HCI /User-Experience Development. http://www.hcii2011.org/index.php?module=webpage&id=47
17. Reinecke, K., Bernstein, A. Tell Me Where You`ve Lived, and I`ll Tell You What You Like: Adapting Interfaces to Cultural Preferences. Proceeding of User Modeling, Adaptation, and Personalization (UMAP), 2009.

18. Shah, H., Nersessian, N. J., Harrold, M. J., Newstetter, W. Studying the influence of culture in global software engineering: thinking in terms of cultural models. 4th international conference on Intercultural Collaboration, ACM, 2012, pp. 77-86.
19. Reinecke, K., Schenkel, S., & Bernstein, A. Modeling a User's Culture. The Handbook of Research in Culturally-Aware Information Technology: Perspectives and Models, IGI Global, 2010.
20. Callahan, E. Cultural similarities and differences in the design of university websites. Journal of Computer-Mediated Communication, 2005, 11(1), 12.
21. Marcus, A., Gould, E. W. Crosscurrents: Cultural Dimensions and Global Web User- Interface Design. ACM Interactions, 2000, 7(4), pp. 32-46.
22. Marcus, A.; Gould, E. W. Cultural Dimensions and Global Web User-Interface Design: What? So What? Now What? Proceedings of Sixth Conference on Human Factors and the Web, Austin, Texas, 2000.
23. Henry, N., Goodell, H., Elmqvist, N., & Fekete, J. D. 20 years of four HCI conferences: A visual exploration. International Journal of Human-Computer Interaction, 2007, 23(3), 239-285.
24. National Council for Scientific and Technological Development (CNPq). Offical website. http://www.cnpq.br/
25. Heimgärtner, R. Reflections on a Model of Culturally Influenced Human–Computer Interaction to Cover Cultural Contexts in HCI Design. International Journal of Human-Computer Interaction, 2013, 29(4), 205-219.
26. Hofstede, G. Cultures and organizations: software of the mind. 2nd ed. New York: McGraw-Hill, 2005.
27. Vatrapu, R; Pérez-Quiñones, M. A. Culture and Usability Evaluation: The Effects of Culture in Structure Interviews. Journal of Usability Studies, 2006, v 1, 4, pp. 156-170.
28. Blanchard, E.,Mizoguchi, R., Lajoie, S. P. Structuring the Cultural Domain with an Upper Ontology of Culture. In E. G. Blanchard and D. Allard (Eds.),

Contextualised Student Modelling for Enculturated Systems

Phaedra Mohammed and Permanand Mohan

Department of Computing and Information Technology, The University of the West Indies,
St. Augustine, Trinidad and Tobago
{phaedra.mohammed@gmail.com, Permanand.Mohan@sta.uwi.edu}

Abstract. Contextual student modeling, also called cultural profiling or cultural modeling, refers to the process of building a computational representation of the cultural identity and background of a student. Previous works have been done that identify and use certain environmental dimensions for such a model. In this paper, a new approach is proposed that uses additional dimensions, and incorporates combinations of dimension clusters to represent and quantify a student's expression of socio-cultural group traits and preferences. The viability of this approach is demonstrated through the use of a prototype that collects dimension data and generates estimates of a student's association with particular socio-cultural groups in five categories. An evaluation of the prototype revealed that estimates were rated as reasonable and acceptable by students and confirms that the approach extends current efforts in the field of culturally-aware tutoring systems for modeling student's cultural context.

Keywords: contextual student model, cultural element, dimensions

1. Introduction

Contextual student modeling, also called cultural profiling or cultural modeling, refers to the process of building a computational representation of the cultural identity and background of a student. This identity is shaped by many dimensions that originate from an individual level such as personal demographics and from a group level such as religious or ethnic influences. The first challenge that arises in contextual student modeling is identifying which dimensions should be modeled, and determining to what extent a dimension affects a student's personality, preferences, and opinions. The second challenge that arises is whether combinations of these dimensions can be worked out such that a student's expression of particular traits and values, shared by a cultural group, are represented and measured relative the group's expression of said traits and values. The final challenge that arises in contextual student modeling is evaluating whether a computational model generated for a student is a reasonable and acceptable representation of the student's particular cultural identity and background.

This paper tackles all three challenges in a systematic manner by looking at culture as a form of context. When culture is looked at as context or rather as a focused collection of metadata, these challenges becomes more tractable and the issues that need to be dealt with start to take on a computational form. The environmental context of an individual is therefore made up of several dimensions of metadata. These contextual dimensions fall into two groups: contextual factors and contextual influences. A contextual factor is something that brings about a particular effect on an individual and can be quantified discretely. A contextual influence is something that brings about a particular effect on an individual but whose exact nature is not readily known and can take on a range of values.

Several key ideas in this paper are based on the works of Blanchard, Mizoguchi, and Lajoie [3] who define the concept of cultural elements and cultural groups. A contextual element is considered to be a type of cultural element. It is an observable manifestation of culture and can be present in educational content expressed as different forms of media (text, pictures, videos, and audio). A contextual group on the other hand is a collection of individuals sharing similar values for contextual dimensions. Contextual groups and individuals are related by these contextual dimensions. The strength of this relationship is determined by the amount of overlap of dimension context and by the individual’s expression of particular dimensions in the intersection. These definitions are central to the approach taken in this paper for dealing with the challenges outlined earlier with the goal of defining a contextual student model.

The paper is organized as follows. Section 2 identifies twenty four dimensions of context for a contextual student model (CSM) based on related research, and explains the rationale for the new dimensions identified in this paper that have not been used in culturally-aware tutoring systems (CATS) research before. Section 3 discusses how these dimensions were clustered based on relevance to particular contextual groups, for the purpose of generating estimates of a student’s level of membership to five contextual groups. Section 4 then describes the ontology-based design of the CSM and the implementation of a rule-based approach for generating contextual estimates. Section 5 outlines experiments that were conducted to evaluate the CSM design and performance together with the results of these experiments. Section 6 gives an analysis of the results and the paper concludes in Section 7 with the future plans for the CSM.

2. Environmental Context: Factors and Influences

Several dimensions have been recurring in the literature as having an effect on students from a cultural perspective. The most common ones include age, gender, nationality, native languages, religion, ethnicity, emotional disposition, and locations of residence and study [4, 5, 6, 8, 9]. Of these characteristics, some are quantifiable and can be considered to be contextual factors such as age, gender, nationality, and locations of residence and study. The remaining traits and qualities such as ethnicity and religion are less easily quantified and are therefore considered to be contextual influences. A good rule of thumb for distinguishing between a factor and an influence is the answer to the following question: For a given characteristic C, how much of a C is the student in question? If the answer can be within a range of potential values then that characteristic is most likely an influence otherwise it is a factor.

Twenty four contextual dimensions have been identified for the CSM based on the works of [1, 5, 9]. The first set of dimensions for the CSM consists of personal factors: age, gender, country of birth, the locale[1] where the student lives, and the schools where the student has studied at primary, secondary and tertiary level. In order to model the historical context of a student, the CSM includes three school-related dimensions that identify locales which would have shaped a student's context over the duration of his/her time in school. The AdaptWeb project [5] uses characteristics similar to the locales of study but their work manipulates IP addresses to identify only one current locale of study for the student. The second set of CSM dimensions consists of personal influences: the student's religion, ethnicity, and native language. Religion influences have been used in [2], language influences have been used in ActiveMath [7] and ethnicity influences have been used in embodied conversational agents [10]. The CSM combines and reasons about the student's context using all three influences since the combination changes the individual impact of a particular influence and can affect the student's perception, interpretation and magnitude of response to a particular contextual element.

The third set of CSM dimensions originate from social units surrounding the student, in this case the student's parents. This is based on the work of Reinecke, Reif, and Bernstein [9] who identified that parents have an impact on users specifically through their language and nationality. The factors in this set include the parents' occupations, their occupation locales, and their ancestral home locales. This kind of context has not yet been used computationally in CATS. The reasons for including these factors stem from the assumptions that students typically visit their parents' workplaces, can be influenced educationally by the kinds of occupations that their parents have, and may frequent the locales where their parents grew up because of existing familial ties to the areas. This leads to the influences in this set which include the parents' religions, ethnicities, native languages, and level of personal influence on the student. The first three are self-explanatory but the strength of their impact depends on the fourth influence. Blanchard [1] discussed the situation of socio-cultural groups affecting the receptivity of individuals to particular cultural elements. The level of personal influence that a parent has on a child affects the child's involvement, beliefs, understanding, and behaviour regarding religion, ethnicity and language. This is therefore an example of socio-cultural group influence at a finer level of granularity and consequently, these dimensions were included in the CSM in order to separate, quantify and structure as best as possible the nature and the strength of control that a parent's context may have on shaping the student's context.

3. Contextual Student Model (CSM) Estimates

The dimensions in the CSM fall into five categories that describe particular contextual groups: geographical groups, religious groups, ethnic groups, groups that share similar education levels, and groups that are familiar with particular physical environment settings and terrains. The CSM generates estimates for each group using a

[1] A locale is considered to be a city, town, village, or hamlet that is officially recognised in a country.

combination of multiple dimensions because individual dimensions have been shown to have limited predictive capabilities when considered in isolation [4].

Geographical estimates are produced using the locale-based dimensions: the locales of the student's residence, parents' ancestral homes, parents' jobs and the student's schools. Two geographical estimates measured as ordinal and cardinal points are produced for the student: a dominant geographic region and a secondary geographic region based on which areas of the country his/her activities most frequently take place. Religious estimates are produced using the religion-based dimensions: the religion of the student, parents, and schools (if any), and the parents' level of influence on the student. Two religious estimates measured as percentages are produced for a student, a dominant religious influence and a secondary religious influence. The dominant influence would be derived from the religious group that student belongs to whereas the secondary influence would be based on the remaining dimensions. A secondary religious influence does not imply that the student belongs to that religious group but rather that the student is aware of that religious group and would have a partial membership because of that awareness. Schools in a country can have either no religious influence if they are non-religious or can influence student knowledge of the norms and practices of a particular religious group if the school is denominational.

Ethnicity estimates are produced using the ethnicity-based dimensions: the ethnicity of the student, parents, and the national ethnicity distributions for the student's residence locale. The distributions are used to approximate the influence on the student of the two largest ethnic groups in his/her locale. Two ethnicity estimates measured as percentages are produced here as well where the dominant ethnicity influence corresponds to the student's ethnicity and the secondary influence would be based on the parent's ethnicities and degree of influence that the parents have on the student. Educational estimates are produced using the schools attended by the student and the national educational statistics for the student's residence locale with the possible values of high, mid-high, mid, mid-low or low. This estimate reflects the level of education of the societal unit in the student's geographical region and does not mean that the student has a low or high level of education. This estimate allows the CSM to gauge how familiar a student would be with different levels of language. Low to mid-low educational estimates imply that more colloquial language would be commonly used by members of society in that particular area compared to more formal language for areas with mid-high to high levels. It is of note to mention that the parents' occupations are suitable factors for this estimate but were not included at this time.

Terrain or setting estimates are produced using the locales of the student's residence, student's secondary school, parents' ancestral homes, parents' jobs, and the parents' level of influence on the student. Three terrain/setting estimates are produced and each estimate may contain one or more categories with percentages of membership. Economic activity context captures whether a student's locale is influenced by industrial, residential, commercial, agricultural or sporting activities. Terrain context captures the type of physical environment the student may be familiar with such as coastal, desert, grassland, mountainous, forested, tundra or wetland terrains. These are based on his/her dominant geographic influences in the country. Urban/rural/semi-rural context deals with the population density of the student's locale. Together, these three areas contribute towards the terrain/setting estimates for a student. Overall, the five categories of estimates are related to the student's contextual identity through specific combinations of contextual dimensions in the CSM and model the degree of a student's membership to a particular contextual group.

4. CSM Design and Implementation

The CSM was implemented using Java and JESS (Java Expert System Shell) and has an ontological design but was implemented using a rule-based approach for prototyping. Figure 1 below shows the main concepts and relationships in the CSM.

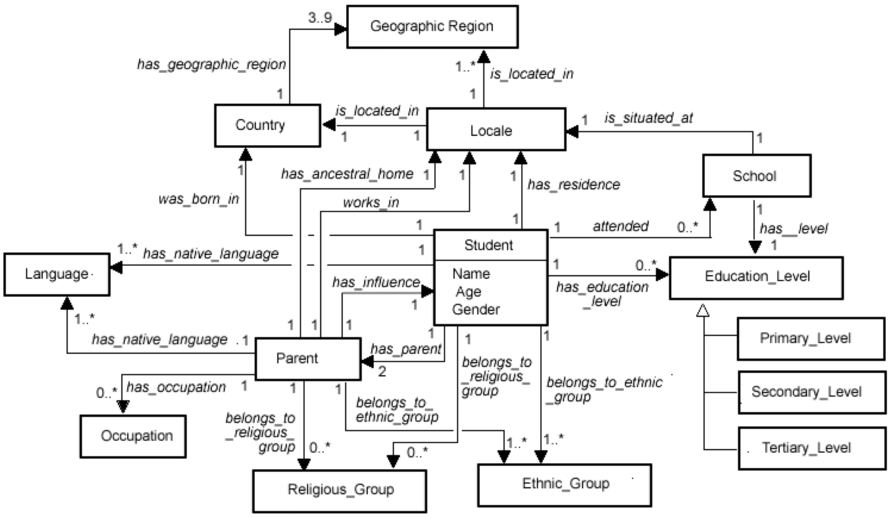


Fig.1. Metadata Structure of the Contextual Student Model

All of the concepts are not shown in the diagram due to space constraints. Each of the twenty four contextual dimensions described in Section 2 are included in the CSM and are supplemented with statistical data from the target country's national statistical office. Data on schools, locales, ethnic groups and their distributions, religious groups and their distributions, population distribution, economic activities across locales, terrain and physical data for locales were loaded into the CSM and used to generate the estimates described in Section 3. Values for the dimensions are sourced from either the student or from the target country's national statistical office. For example, the values for locale would be selected from the list of locales situated in the target country recorded by the national statistical office for the country. Similarly, the value for religion would be selected from the list of religious groups common in the target country as recorded by the statistical office. The use of country-level data to define the value spaces for some of the dimensions allows subtle nuances and variations in naming conventions for these values to be considered. Furthermore, compared to asking the opinion of a few members of a target country, the national records provide a more comprehensive, objective snapshot of the possible values that a dimension can take.

The research in this paper builds upon the approach elaborated in [2] for quantifying a student's membership to a contextual group. Blanchard [2] measured this relationship as a membership score dynamically calculated as the weighted difference between the student's characteristics and those of a contextual group. Our approach also uses weighted values but differs in the calculation of the membership score and the determination of weights. The weights in our approach are applied to contextual influences and are based on two sources of data: parent's level of influence and country level statistical data. This improves upon the approach in [2] by using weights directly related to the student's context. This means that the CSM would strengthen one student's contextual group membership for a particular category and weaken the same membership for another student as their weights change based on the significance of a dimension for their particular cluster context. If two students have similar contexts but different parental influences for example then their estimates would vary. The same holds true for different statistical distributions for the dominant influences in their contextual categories. In this case, further information is derived from a dimension using statistical data from the central or national statistical office in the country where the students reside for the course of their studies. In doing so, the socio-cultural group contexts of the social units relevant for the students are factored into the estimates. These two features advance the calculation described in [2]. Furthermore, the definition of groups that relate to contextual dimensions and elements in this paper extend content manipulation beyond the educational dimensions used in [2].

There are several potential uses envisioned for the CSM, and these hinge on adaptation at the application layer of CATS environments. One use could involve the dynamic selection of contextual elements deemed suitable for adapting learning content based on the values and estimates in the CSM. Here, the contextual elements that appeal most to students could be inserted into educational content thereby producing contextualised content. Another use of the CSM could involve the generation of contextualised instructional feedback with emotive qualities. Affective feedback generated using casual or formal varieties of language as defined by the CSM could be used to elicit different emotive responses in students in accordance with instructional goals.

5. CSM Evaluation and Results

Two studies were conducted in response to the research challenges posed at the beginning of the paper using the CSM. The first study evaluated the likelihood that the data required for generating a contextual student model will be readily supplied by users. The second study evaluated the acceptability of the estimates produced by a CSM application, built for the context of Trinidad and Tobago, based on student ratings of the estimates. This section describes the methods and results of each study.

5.1 Likelihood of Data Collection for the CSM

An online questionnaire was administered to thirty six participants (36) from a cross section of the population in Trinidad. It consisted of questions dealing with a participant's willingness to supply information on a contextual dimension. Participants were asked to answer whether they would be willing to supply information, uncomfortable but willing to supply information, or unwilling to supply information for each of the twenty four dimensions in the CSM. Figure 2 shows the number of responses categorised by user willingness and comfort to supply contextual data. Out of 864 responses, 786 responses were classified as willing and comfortable (91%), 49 responses were classified as willing but uncomfortable (5.7%) and 29 responses were classified as unwilling (3.3%). Overall, the majority of users were willing and comfortable to supply contextual data on themselves and their social units (parents).

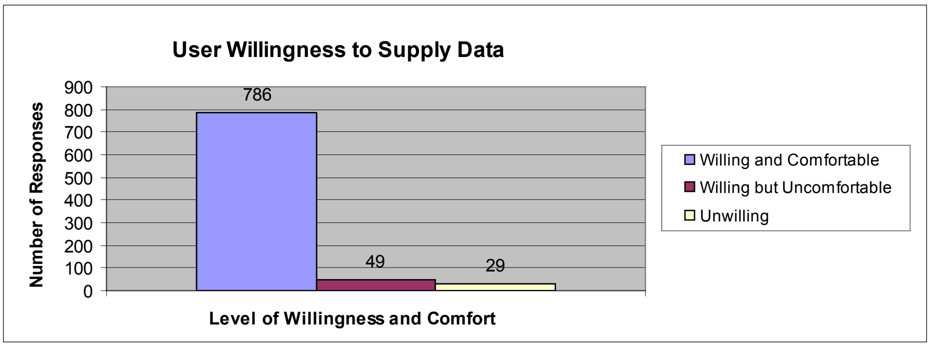
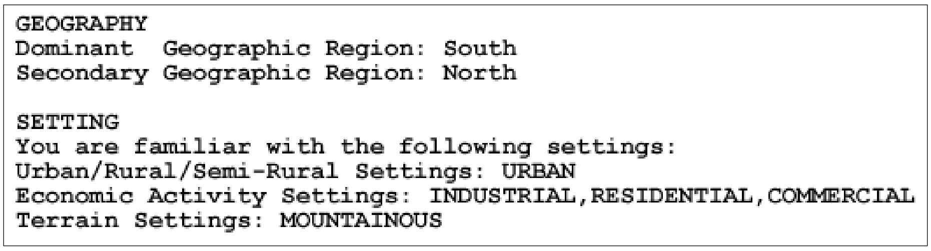


Fig.2. User Willingness to Supply Data for a Contextual Student Model

5.2 Acceptability of Contextual Estimates Generated by the CSM

Thirty (30) undergraduate students enrolled in a programming course at UWI voluntarily participated in the experiment. The students ran the CSM application which prompted for data for each of the twenty four factors. Using this data, the CSM application produced estimates of contextual influences in the following areas: geography, religion, ethnicity, education, and physical setting. Students were asked to rate the estimates for correctness using a four point Likert scale rating. Usage logs were stored and retrieved from a server for analysis.



```
GEOGRAPHY
Dominant  Geographic Region: South
Secondary Geographic Region: North

SETTING
You are familiar with the following settings:
Urban/Rural/Semi-Rural Settings: URBAN
Economic Activity Settings: INDUSTRIAL,RESIDENTIAL,COMMERCIAL
Terrain Settings: MOUNTAINOUS
```

Fig.3. Sample of CSM Estimates Generated for a Student

Figure 3 shows a sample of the geographic and the terrain/setting estimates generated for a student who lives in an industrialized hilly city in the southern part of Trinidad. The student rated the setting estimate as correct but rated the geographical estimate as mostly wrong even though one of his parents' ancestral homes and work location were situated in the north of the country. The graph in Figure 4 below shows the relative differences in student ratings of the accuracy of the contextual student model estimates that were produced. When ranked in order of increasing accuracy as being either correct or mostly correct the categories are as follows: setting (80%), religion (87%), geography (90%), ethnicity (93.3%), and education (96.7%). The most inaccurate estimates (wrong and mostly wrong) were in the setting category (20%) followed by the religion category (13.3%), and then the geography category (10%). All categories of estimates were rated on average as correct or mostly correct by over 80% of the students. Collectively the estimates were rated as being 89.3% accurate and 10.7% inaccurate.

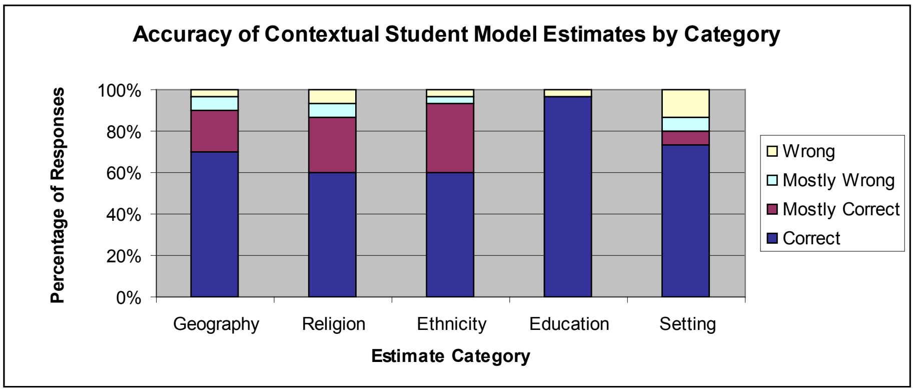


Fig.4. Accuracy of Contextual Student Model Estimates by Category

6. Analysis and Discussion of Results

The first experiment aimed to evaluate the likelihood that the data required for generating a contextual student model will be readily supplied by users. The results showed that the majority of users polled for this experiment were willing and comfortable to supply contextual data about themselves and their parents. Closer examination revealed that all of the users were willing and comfortable to give information about their schools, and languages spoken by themselves and their parents. There were differences in the number of users (ranging from 100% to 77.8%) who were willing and comfortable to supply data for the remainder of dimensions. Users were the least comfortable to give information about their parents compared to themselves but were willing to give levels of influence. Overall, the experiment indicates that users would readily supply information for the majority of dimensions that are used in the CSM to generate the estimates. In the cases where information would not be supplied for particular dimensions such as for parental social units, the information provided for personal dimensions seem to be sufficient for estimating missing data through averages using country-level statistics. It is therefore not unreasonable to conclude that data collection for the CSM is viable for the country investigated. Further study is required to determine whether data collection is viable for other countries since there are differences across countries with respect to what users may be willing to divulge about themselves along with legal and ethical issues as evidenced by the study in [1].

Given that data can be collected for the CSM from users in general, the second experiment aimed to evaluate the acceptability of the estimates produced by the CSM based on student ratings of the estimates. Students were used in this experiment since the intended use of the CSM is for educational purposes. The results showed there were variations in the accuracy ratings for each category of estimates but overall more than 80% of students rated the estimates generated as correct or mostly correct. The setting category was rated as least accurate. This happened possibly because of the limited metadata on the country locations which did not sufficiently distinguish cities or towns as rural compared to semi-rural or even urban for the students. This highlights one limitation of the CSM in depending on statistical data from a country's central statistical office or department. Errors can be introduced into the estimates if the data is incomplete or not specific enough. Nonetheless, the estimate was still reasonably accurate since it was rated as wrong by 13.3% of the students but only mostly wrong by 6.7% of the students. Estimates in the religion category may have fallen short by not assigning a larger weight to the student's religion since a few estimates recorded a different dominant religious factor for students whose religions differed from their parents. Even so, the estimate was still reasonably accurate since it was rated as wrong by 6.7% of the students but only mostly wrong by 6.7% of the students. The estimates for geography, ethnicity and education were rated as over 90% accurate and this shows that these estimates were on point for the students. Despite the accuracy of the estimates, there were cases of students rating the estimates as inaccurate as shown in Figure 5 even though the reasoning for the estimate was logical and made sense for the student's context. Overall the CSM rules, dimension combinations and weightings were reasonable for estimating the student's membership to various contextual groups as indicated by the favorable accuracy ratings.

7. Conclusion and Future Research

The contributions of this paper are the identification of the main contextual dimensions of a student's cultural background that are important for adaptation at the application layer in CATS together with the dimension combinations that work to generate reasonable estimates of a student's membership to various cultural groups. Rules were developed to estimate a student's degree of membership to these contextual groups. Results from the evaluations of the CSM revealed that the model was accurate in assigning contextual group membership scores to students. The techniques described in this paper are non-trivial and harness many pieces of metadata in order to create a reasonable computational representation of a student's contextual background. In doing so, this research has revealed that a considerable amount of effort will be required by practitioners seeking to create contextual student models due to the heavy reliance on model values at a student level, resource level and country level. The CSM approach was developed with generalization at the core since it is important for others to be able to replicate these results in their own country and context in order for CATS research to continue to move ahead. Strategies for building models of student context would be worth very little if the students agree with the model but do not wish to have their cultural context factored into their learning experience.

Future research includes the transition of the CSM prototype to an ontological representation to facilitate reuse and better context matching through ontological alignment and merging with resource contexts. Additional dimensions of personal student contexts will be included in the CSM together with more integrated learner context in order to fine-tune the estimates generated. More importantly, work is planned for the investigation of techniques that allow students to accept, adjust or even turn off contextualisations in culturally-aware tutoring systems.

References

1. Blanchard, E.G.: Is it adequate to model the socio-cultural dimension of e-learners by informing a fixed set of personal criteria? In Proc. 12th IEEE International Conference on Advanced Learning Technologies. 388-392. USA: IEEE Computer Society. (2012)
2. Blanchard, E.G.: Adaptation-oriented culturally-aware tutoring systems: When adaptive instructional technologies meet intercultural education. In Song, H., Kidd, T. (Eds.): Handbook of Research on Human Performance and Instructional Technology. Hershey, PA: IGI Global. 413-430. (2009)
3. Blanchard, E.G., Mizoguchi, R., Lajoie, S. P.: Structuring the cultural domain with an upper ontology of culture. In Blanchard, E., Allard, D. (Eds.): The Handbook of Research on Culturally-Aware Information Technology: Perspectives and Models. 179-212. Hershey, PA: IGI Global. (2011)
4. Blanchard, E. G., Roy, M., Lajoie, S. P., Frasson, C.: An evaluation of socio-cultural data for predicting attitudinal tendencies**.** In Proc. 14th International Conference on Artificial Intelligence in Education, Brighton, UK. 399-406. Amsterdam: IOS Press. (2009)
5. Gasparini, I., Pimenta, M.S., de Oliveira, J.P.: How to apply context-awareness in an adaptive e-learning environment to improve personalization capabilities? In Proc. 30th International Conference of the Chilean Computer Society, SCCC 2011, Chile. 161-170. (2011)
6. Horton, W.: Graphics: The not quite universal language. In Aykin, N. (Ed.): Usability and Internationalisation of Information Technology. 157-187. Mahwah, NJ: Lawrence Erlbaum Associates. (2005)
7. Melis, E., Goguadze, G., Libbrecht, P., Ullrich, C.: Culturally-aware mathematics education technology. In Blanchard, E., Allard, D. (Eds.): The Handbook of Research on Culturally-Aware Information Technology: Perspectives and Models. 543-557. Hershey, PA: IGI Global. (2011)
8. Rehm, M.: Developing enculturated agents: Pitfalls and strategies. In Blanchard, E., Allard, D. (Eds.): The Handbook of Research on Culturally-Aware Information Technology: Perspectives and Models. 362-386. Hershey, PA: IGI Global. (2011)
9. Reinecke, K., Reif, G., Bernstein, A.: Cultural user modeling with CUMO: An approach to overcome the personalization bootstrapping problem. In Proc. First International Workshop on Cultural Heritage on the Semantic Web at the 6th ISWC 2007. 83-90. (2007)
10. Cassell, J.:Social practice: Becoming enculturated in human-computer interaction. In Stephanidis ,C. (Ed.): Universal Access in HCI, Part III, HCI 2009, LNCS 5616. 303-313. (2009)

A Virtual Space for Children to Meet and Practice Chinese

Mei Si

Rensselaer Polytechnic Institute
sim@rpi.edu

Abstract. Second language acquisition after the students have learned their first language is a unique process. One key difference between learning a foreign language and one's mother tongue is that second language learning is often heavily facilitated with digital media, and in particular, through interacting with computers. This project is aimed at leveraging computer game technologies and Microsoft Kinect camera to create engaging and affordable virtual environments for children to "virtually" meet and practice their language and culture skills. We present a uniquely immersive and narrative-based environment for children to meet online and practice Mandarin Chinese with each other, providing a platform that is at once affordable, engaging to students and attractive to teachers.

Keywords: Language and Culture Learning; Virtual Environment; Kinect

1 Introduction

Language acquisition plays an important role in children's cognitive and social development processes. Being able to understand another language and culture is not only fun and useful, but also may facilitate the children's cognitive development.

Second language education is traditionally accomplished through classroom instruction and human tutors. While this is usually an effective approach, it has several drawbacks which limit its application, particularly with a younger audience. Learning a language requires tremendous amount of practice and repetition. Therefore, students are strongly encouraged to practice outside normal classroom hours. However, being able to find appropriate study partners, coordinate a convenient time and location is often difficult, and this is especially true for children. Even when the students can find someone to practice with, due to the students' insufficient language abilities, the practice session can be very frustrating for both the student and the partner, who may be a native speaker. This often causes the students to abort the practice. The students also may not be able to retain the lessons during these practices, because of information overflow.

Computer and network based learning systems on the other hand can keep records of the practice history, and allow people to practice with partners online. Such systems therefore can potentially make the practice process more accessible and effective.

In this project, we are interested in investigating how to create effective second language practice environments. We have created multiple virtual environments for children to "virtually" meet and practice their language and culture skills through leveraging computer game technologies and Microsoft Kinect camera. Among all the foreign languages for children in the US, Mandarin Chinese is by far receiving the most growth of interest over the past years, and therefore is chosen as the first language to be practiced in our games. We present the details of this system in this paper.

Furthermore, we are interested in studying the impact of various factors on students' language practices, and in particular the use of narrative and body movements. Narrative, of course is an integral part of people's lives. It is an important way for people to entertain, as well as to learn about a new society or culture. Its engaging power has been well observed in various media forms, e.g. novels, movies, and dramas. Simulating real-world or fictional scenarios offers a way to practice language in context, and provides the users with motivations and focus.

Moreover, we hypothesize that body movements and gestures can contribute to language learning. Movement, in essence, is a form of thinking. The theories of embodied cognition argue that our body, mind and the environment are tightly integrated, and our decision-making processes, perception and even memory are deeply rooted in our body and bodily movements (Clark, 2008). Gesturing is a perfect example. Expressive gestures are an important aspect of language use and communication. On the other hand, spontaneous gesturing, which do not directly relate to language use, has been shown for facilitating learning and recall of abstract concepts (Goldin-Meadow, 2003). This is because memory can be off loaded to body-environment relationships that are "artificially" created by us. In the future work section, we lay out the experimental studies we plan to conduct for evaluating the effects of these two factors.

2 Related Work

With the rapid development of computer and game console hardware, graphics, artificial intelligence and network technologies in recent years, computer aided pedagogical systems and intelligent pedagogical agents have been widely used for tutoring and training purposes, ranging from math (Beal, Walles, Arroyo and Woolf, 2007) and physics tutoring (Ventura, Franchescetti, Pennumatsa, Graesser, Jackson, Hu, Cai and the Tutoring Research Group, 2004) to language and social skill training (Johnson, Marsella, Mote, Si, Vilhjalmsson and Wu, 2004; Traum, Swartout, Marsella, Gratch, 2005), and from life style suggestions (Zhang, Banerjee and Luciano, 2010) to PTSD (Rizzo, Newman, Parsons, Reger, Difede, Rothbaum, Mclay, Holloway, Graap, Newman, Spitalnick, Bordnick, Johnston and Gahm, 2009) and Autism interventions (Boujarwah, Riedl, Abowd and Arriaga, 2011).

Similarly, the effective use of language training has been demonstrated in immersive virtual environments such as the Tactical Language Training System (Johnson, Marsella, Mote, Si, Vilhjalmsson and Wu, 2004), and Rosetta Stone. O'Brien, Levy, and Orich describe a CAVE-based language learning environment targeted at more general L2 applications, in which students explore a virtual model of Vienna in search of the mayor's missing daughter (O'Brien, Levy, and Orich, 2009). Chang, Lee and Si have investigated using immersive narrative and mixed reality for teaching Mandarin Chinese to college students (Chang, Lee and Si, 2012).

Language and culture training involves more than teaching the students how to speak. In real life, people use non-verbal behaviors -- gazes, gestures, and body movements -- to accompany their speech. Not all of the non-verbal behaviors are straightforward to mimic for foreigners. Most existing virtual training environments require the learner to sit in front of a computer and use keyboard and mouse to interact. The learner therefore cannot practice their non-verbal behaviors and conversational skills at the same time.

Gesture based natural user interfaces has been explored in cultural training (Rehm, Leichtenstern, Plomer, and Wiedemann, 2010; Kistler, Endrass, Damian, Dang and André, 2012). In this project we combine gesture based user interface with narratives and puzzles to provide the users with a platform to practice their verbal and non-verbal skills together. More specifically, we created three types of Kinect enabled virtual environments. All of these virtual environments allow multiple users to log in from different locations. The users can control a character's body movements using a Kinect camera and simultaneously have voice chat with other users. In fact, they have to discuss and collaborate with each other to solve the problems presented in the virtual environments. We hope these virtual environments can thus engage the users and help the users to practice in a natural way.

3 Project Description

This project is aimed at leveraging computer game technologies and Microsoft Kinect camera to create engaging and affordable virtual environments for children to "virtually" meet and practice their language and culture skills in Mandarin Chinese. Our goal is to create affordable, engaging and realistic learning environments for children to meet and practice Chinese. This project is not aimed at replacing language classes or human tutors, but is meant to supplement classroom instruction.

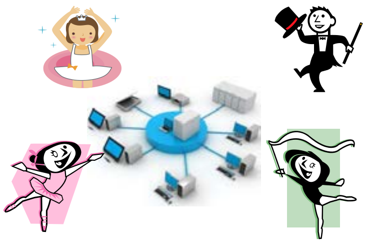


Fig. 1. System Architecture

3.1 Overview of the System

Figure 1 illustrates the overall architecture of the system. This system contains a central server and multiple clients. The server handles most of the computation, and therefore low-end machines can be used as clients. Each client needs to be equipped with a Kinect camera, microphone and speaker. The Kinect camera enables the system to map the user's body movements to a character's body movements, and thus allow the user to directly control the character's motions using his/her body. We choose to use the Kinect camera in this project, because it provides this important functionality, and is affordable, non-invasive and easy to install, which are important features when considering deploying the project outside of the lab.

3.2 Development Environment

The Unity game engine was used for developing this project. Unity supports the development of multi-player network games, and can easily produce the final executable for different platforms. Unity is also easily compatible with the OPENNI package for driving the Kinect camera and Teamspeak 3 package for providing real time multi-user online chat, which will be described in more details below. As a result, we expect minimal effort for deploying the final project at interested schools.

We want to seamlessly integrate the voice chat function into the rest of the application, so that the users do not need to perform any special operations to chat with each other. Considering the age group of our users, it is very important to make all the interactions with the system feel natural. Moreover, we want to be able to replay the whole game session for research purposes, and for the children's teachers and parents. In order to add a voice chat component to our system, we used Teamspeak 3, a popular voice chat service for gaming and other consumer uses that provides both off-the-shelf and SDK tools, to create a voice chat server that we can host in-house and a client that is integrated into the client side of the system. This means that upon entering the environment, the user is immediately connected to the voice chat server with an open microphone so that all parties can begin talking to each other right away. It also means that all the voice chat messages come through our server so we can keep record of them.

3.3 Environments and Characters

Three types of virtual environments were created, which are described below in the order of the amount of background stories involved.

Cao Chung Weighing an Elephant

We have created a 2.5D virtual environment (Figure2) which is based on a traditional Chinese children's story – Cao Chong Weighing an Elephant:

> *This happened about one thousand and seven hundred years ago. One day somebody sent Cao Cao, the king of WEI, an elephant. Cao Cao wanted to know its weight. "Who can think of a way to weigh it?" He asked. But*

nobody knew what to do, because there was nothing big enough to weigh it. Then Cao Chung, one of the king's young sons, came up and said, "Father, I've got an idea. Let me have a big boat and a lot of heavy stones, and I'll be able to find out the weight of the elephant." Cao Cao was surprised, but he told his men to do as the boy asked.

When the boat was ready, the boy told a man to lead the elephant down into it. The elephant was very heavy, and the water came up very high along the boat's sides. Cao Chung made a mark along the water line. After that the man drove the elephant onto the bank. Cao Chong then told the men to put heavy stones into the boat until the water again came up to the line. Cao Chung then told the men to take the stones off the boat and weigh them one by one. He wrote down the weight of each stone and then added up all the weights. In this way he got the weight of the elephant.

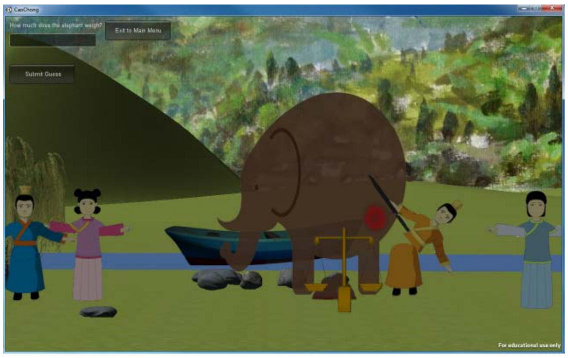


Fig. 2. Cao Chung Weighing an Elephant

In this virtual world, the users are asked to play the kids in the story and to find the right way to weigh an elephant without hurting it. The users are provided with multiple tools, such as a knife, which is not big enough to chop the elephant into pieces but can wound the elephant, a scale that is not large enough to weigh the whole elephant and multiple stones as in the original story. When each user enters the game, they are provided with different sets of information regarding where the tools are. The users can find and try out the tools. They are encouraged to discuss how to use the tools and how to solve the problem with each other.

The characters were modeled in 2D with movable body parts. Using a Kinect camera, the user can control the characters’ movements through their own body movements. The characters mimic the user’s actions, e.g. the user can move around, wave his/her hand, and bend down to pick up or drop an object.

The Elephant and the Blind Men

The second practice scenario is created based on another traditional Chinese children’s story – The Elephant and the Blind Men:

> *Once upon a time, an elephant came to a small town. People had read and heard of elephants but no one in the town had ever seen one. Thus, a huge crowd gathered around the elephant, and it was an occasion for great fun, especially for the children. Five blind men also lived in that town, and consequently, they also heard about the elephant. They had never seen an elephant before, and were eager to find out about elephant. Then, someone suggested that they could go and feel the elephant with their hands. They could then get an idea of what an elephant looked like. The five blind men went to the center of the town where all the people made room for them to touch the elephant. Later on, they sat down and began to discuss their experiences. One blind man, who had touched the trunk of the elephant, said that the elephant must be like a thick tree branch. Another who touched the tail said the elephant probably looked like a snake or rope. ... Finally, they decided to go to the wise man of the village and ask him who was correct. The wise man said, "Each one of you is correct; and each one of you is wrong. Because each one of you had only touched a part of the elephant's body. Thus you only have a partial view of the animal. If you put your partial views together, you will get an idea of what an elephant looks like."*

Just like in the original story, in this practice scenario, each user can only see a portion of a large object, and they have to discuss with each other to figure out what the object is. In addition to the elephant in the original story, we are also showing other 2D and 3D objects with different levels of difficulties for this practice. Figure 3 shows an example. Using the Kinect camera, the users can use their hands to move their camera view of the object a little bit to see more of the object. The user will never be able to see the whole object. We designed this function to allow us to later evaluate whether encouraging body movements will engage the users more in a learning environment like this.

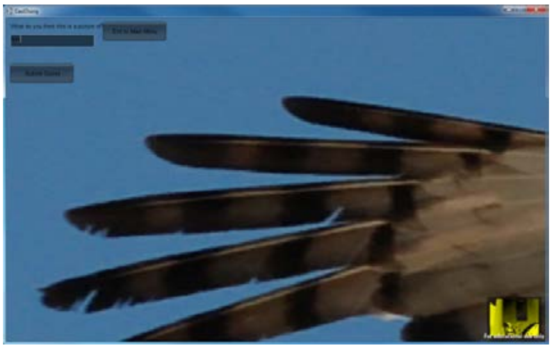


Fig. 3. The Elephant and the Blind Men

Chat Rooms

Finally, we have created two 3D environments with 3D characters in them, as shown in Figures 4 and 5 below. One is a student café with realistic models of human characters. The other is a living room with children characters modeled in cartoon style. They were designed to attract users of different ages. In both environments, the users can control a character, “walk” around in the environment, talk with other characters, and interact with objects in the room. For example, in the café, the user can pick up a cup and hand it to a virtual characters or to another user. The user can also collaborate with another user and push the tables around.

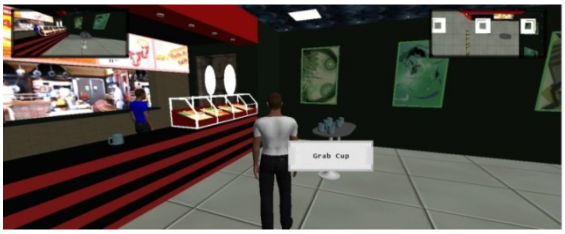


Fig. 4. School Café

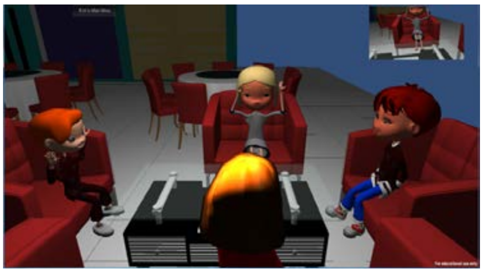


Fig. 5. Living Room

4 Current Status and Planned Evaluation

We have finished implementing the system and are currently conducting informal usability testing. We are planning on two formal evaluations in the future.

First, the goal of this project is to supplement classroom instruction and exercise, raise and sustain children’s interest in learning Chinese language and culture, and help them practice their language skills outside of classroom. The overall effectiveness of the project will be evaluated by comparing the learning outcomes from children who regularly use this project and those who do not use this project. We will be using the school’s standard assessment for evaluating the students’ performances.

Secondly, we want to evaluate the effects of using narrative and body movements in language learning. For this purpose, we have also created a mouse and keyboard version of the system. The same menu interface is used. However, the user control his/her character’s body movements using mouse and keyboard instead of using their own body movements. The three types of virtual environments we developed for this project involve different amount of narrative components. We hypothesize that the environments with more narrative components can engage the users more, and using body movements with a Kinect camera is a more engaging/effective learning approach compared to using mouse and keyboard.

We will use a 3*2 between-group design with the amount of narrative involved and the type of user interface as two independent variables. Each subject will attend a 3-session study. They will spend one hour in the lab interacting with our system on three consequent days. The course of their interactions will be recorded. On the last day, their learning results will be evaluated. The specific measurements for evaluation will be determined together with our teaching consultants when we develop our learning materials. Based on the results from this experiment, we will conduct a second study exploring ways to encourage the desirable behavior patterns in the students.

5 Conclusion

Learning a language requires a tremendous amount of practice both inside and outside of the classroom. One common problem faced by language learners is where/how to find people to practice with and what to talk about with strangers. This is especially true for children because they have to rely on their parents or other adults for transportation.

In this project we propose to attack these problems by creating virtual spaces with narratives and puzzles embedded in them. We created three types of Kinect enabled virtual environments. The success of this project will make finding and meeting practice partners a lot easier, in addition to the numerous benefits a computer based pedagogical system can provide, such as automatically keeping a record of one's practice history.

The rapid development of computer technologies in recent years enabled a variety of user interfaces to be continently accessible in people's everyday life, ranging from the traditional mouse and keyboards interface to touch screens and touch free camera based technologies for interactions. This work provide a platform for studying how the form of interaction affects children and young adults' language learning processes, and how the design of the learning systems can leverage this effect and make the students' learning process more effective.

Reference

1. Beal, C. R., Walles, R., Arroyo, I., & Woolf, B. P.: On-line tutoring for math achievement testing: A controlled evaluation. *Journal of Interactive Online Learning*, 6 (1), 43-55, 2007.
2. Boujarwah, F., Riedl, M. O., Abowd, G. and Arriaga, R.: REACT: Intelligent Authoring of Social Skills Instructional Modules for Adolescents with High-Functioning Autism. *ACM SIGACCESS Newsletter*, vol. 99, 2011.

3. Chang, B., Sheldon, L. and Si, M.: Foreign language learning in immersive virtual environments. In Proceedings of *IS&T/SPIE Electronic Imaging*, Burlingame, CA, 2012.
4. Goldin-Meadow, S.: Thought before language: Do we think ergative? In D. Gentner & S. Goldin-Meadow (Eds.), *Language in mind: Advances in the study of language and thought*, pp. 493-522. Cambridge, MA: MIT Press, 2003.
5. Johnson, W. L., Marsella, S.C., Mote, N., Si, M., Vilhjalmsson, H., and Wu, S.: Balanced Perception and Action in the Tactical Language Training System. In proceedings of *Balanced Perception and Action in ECAs in conjunction with AAMAS*, July 19-20, New York, 2004.
6. Kistler, F., Endrass, B., Damian, I., Dang, C.T. and André, E.: Natural interaction with culturally adaptive virtual characters. *Journal on Multimodal User Interfaces,* 6 (1-2), pp. 39-47, 2012.
7. Miller, L.C., Appleby, R. P., Christensen, J.L., Godoy, C., Corsbie-Massay, C., Read, S. J., Marsella, S., and Si, M.: Virtual agents and virtual sexual decision-making: Interventions for on-line applications that change real-life risky sexual choices. In S. Noar & N. Harrington (Eds.), *Interactive Health Communication Technologies: Promising Strategies for Health Behavior Change*. Mahwah NJ: Lawrence Earlbaum Associates, 2011.
8. O'Brien, M., Levy R. & Orich, A.: Virtual Immersion: The Role of CAVE and PC Technology, *CALICO Journal,* 26 (2), 2009.
9. Rehm, M., Leichtenstern, K., Plomer, J. and Wiedemann, C.: Gesture activated mobile edutainment (GAME): intercultural training of nonverbal behavior with mobile phones. In proceeding of *the 9th International Conference on Mobile and Ubiquitous Multimedia*, 2010.
10. Rizzo, A., Newman, B., Parsons, T., Reger, G., Difede, J., Rothbaum, B.O., Mclay, R.N., Holloway, K., Graap, K., Newman, B., Spitalnick, J., Bordnick, P., Johnston, S. and Gahm G.: Development and Clinical Results from the Virtual Iraq Exposure Therapy Application for PTSD. In proceedings of *IEEE Explore: Virtual Rehabilitation*, Haifa, Israel, 2009.
11. Traum, D., Swartout, W., Marsella, S., and Gratch, J.: Fight, Flight, or Negotiate: Believable Strategies for Conversing under Crisis, In proceedings of *the 5th International Conference on Interactive Virtual Agents*, Kos, Greece, 2005.
12. Ventura, M.J., Franchescetti, D.R., Pennumatsa, P., Graesser, A.C., Jackson, G.T., Hu, X., Cai, Z., and the Tutoring Research Group.: Combining Computational Models of Short Essay Grading for Conceptual Physics Problems. In J.C. Lester, R.M. Vicari, & F. Paraguacu (Eds.), *Intelligent Tutoring Systems 2004*, pp. 423-431. Berlin, Germany: Springer, 2004.
13. Zhang, S., Banerjee, P.P., Luciano, C.: Virtual exercise environment for promoting active lifestyle for people with lower body disabilities. In Proceedings of *the 2010 International Conference on Networking, Sensing and Control (ICNSC),* pp. 80-84, Chicago, IL, 2010.

A Synergic Neuro-Fuzzy Evaluation System in Cultural Intelligence

Zhao Xin WU [1], Jacqueline BOURDEAU [2], Roger NKAMBOU[3]

[1,3] Computer Science Department, University of Quebec in Montreal
PO Box 8888, Downtown, Montreal, QC, H3C 3P8, Canada
wu.zhao_xin@courrier.uqam.ca; nkambou.roger@uqam.ca
[2] LICEF, Télé-Université, 100 Sherbrooke, O., Montreal, QC, H2X 3P2, Canada
Jacqueline.bourdeau@licef.ca

Abstract. In today's age of globalization, cultural awareness has become a challenge for designers of tutoring systems to include the cultural dimension in the tutoring strategy and in the learning environment. Nevertheless, cultural awareness is also a domain to be learned by a student, and a competency that can be assessed. Research on cultural intelligence has provided a new perspective and presented a new way to alleviate issues arising from cross-cultural education. To date, no research on cultural intelligence has been empirically computerized with soft-computing technology. This research aims to invent a cultural intelligence computational model and to implement the model in an expert system through the use of artificial intelligence technology. The purpose of this study is to provide intercultural training for individuals to solve the intercultural adaptation problems they may be faced with in a variety of authentic cross-cultural situations.

Keywords: Cultural Intelligence, Fuzzy Logic, Artificial Neural Network, Expert System, Hybrid Technology

1 Introduction

We live in an era of globalization where international activities between different cultures and intercultural communications and exchanges are becoming more common and are taking on much greater importance than ever before. Cultural awareness has become a challenge for designers of tutoring systems to include the cultural dimension in the tutoring strategy and in the learning environment. Nevertheless, cultural awareness is also a domain to be learned by a student, and a competency that can be assessed. Culture is an ill-defined domain [1]. Culture can play a significant role in the success or failure of face-to face encounters [2, 3], and because of cultural diversity, "*Culture is more often a source of conflict than of synergy. Cultural differences are a nuisance at best and often a disaster*" (Dr. Geert Hofstede). Moreover, cultural knowledge is generally represented by natural language, in ambiguous terms, and it is difficult for traditional computing techniques to cope with these. In such a context, globalization and traditional computing techniques have encountered two major challenges: the first is, for human beings, how to adapt to cultural diversity, and the second is, for computers, the processing of soft data and the representation of human-like thinking. In the field of Culturally-Aware Tutoring Systems (CATS), several efforts have been conducted towards a declarative knowledge representation of culture as a phenomenon in order to foster and assess the awareness of cultural differences among human beings, and of their impact on behaviour and attitudes [1, 2, 3, 4]. The problem addressed in this paper is not how learning environments can adapt to culture, but how to assess human beings in terms of their level of cultural awareness, and make recommendations for their training.

We became interested in the research on cultural intelligence, which provides a new perspective and a new way to alleviate cultural issues that arise in globalized environment. Following Earley and Ang [4], Cultural Intelligence is thereafter called Cultural Quotient (CQ). The higher the CQ that people possess, the more effective their performance and adjustment will be in culturally diverse settings [5]. CQ can also be improved by training the people involved in such settings. The most important point to consider is how to precisely evaluate CQ and provide relevant suggestions to improve it. However, current studies on CQ have used traditional methods to measure users' CQ and have relied primarily on questionnaires to find solutions to CQ problems traditionally confined to the work of culture experts and researchers. The best way to enable non-expert users to make use of CQ knowledge at the present time is to computerize CQ. A great deal of CQ knowledge, however, is expressed as 'fuzzy data'. Dealing effectively with these is beyond the scope of traditional computer technique. Research on CQ has never been empirically computerized to date. Additionally, in reference to cultural aware intelligent systems, researches concerning the Artificial Neural Network (ANN) and fuzzy logic technologies to CQ have not been used before. Up until now, application of this soft-computing technology to CQ has not been found in literature reviews.

This research attempts to provide effective solutions for the above-mentioned problems. Based on advanced AI technologies, a CQ computational model is invented and implemented in an expert system. This system has successfully manipulated linguistic variables, soft data and human-like reasoning.

2 What is Cultural Intelligence?

The definition of CQ relies upon an understanding and an interpretation of a definition of 'culture' itself. According to Hofstede [6], culture is '*The collec-*

tive programming of the mind which distinguishes the members of a human group from another'. Sperber claims that culture can be understood as an epidemiology of representations [7], Kroeber and Kluckhohn [8], in their article 'Culture: A Critical Review of Concepts and Definitions', inventoried a list of over 200 different definitions for the word 'culture'. Moreover, when referring to someone's ability to understand and adapt to different cultures, some authors use the term '**Inter- cultural Sensitivity**' [9]. We adopted the definition proposed by Earley and Ang [4], who define CQ as the ability to collect and process information, to form judgments, and to implement effective measures in order to adapt to a new cultural context. Earley and Mosakowski [10] define CQ as a complementary intelligence form which may explain the capacity to adapt and face diversity, as well as the ability to operate in a new cultural setting. Earley and Mosakowski stress that people with a relatively high CQ level often appear at ease in new situations. They understand the subtleties of different cultures, so they can avoid or resolve conflicts early. Peterson interprets CQ in terms of its operation [11]. He believes that, for the concept of CQ, the definition of culture is compatible with the cultural values of Hofstede. Peterson also describes CQ as the communicative capabilities which improve working environments. In other words, all workers have the ability to communicate efficiently with customers, partners and colleagues from different countries in order to maintain harmonious relationships. Brisling et al. define CQ as the level of success that people have when adapting to another culture [12]. Thomas describes CQ as the capability to interact efficiently with people who are culturally different [13]. Johnson et al. define CQ as the effectiveness of an individual to integrate a set of knowledge, skills and personal qualities so as to work successfully with people from different cultures, both at home and abroad [14]. Finally, Ang et al. [15] define CQ as the conceptualization of a particular form of intelligence based on the ability of an individual to reason correctly in situations characterized by cultural diversity. Ang and Van Dyne [18] paid special attention to how a culturally diverse environment works. They refined the concept of Earley et al. [4] to consist of four dimensions of CQ: metacognition, cognition, motivation and behavior. This structure has been widely used in the following cultural research and studies.

3 Data and Knowledge Acquisition in the Application Domain

We collected data and CQ knowledge by reviewing books, documents, manuals, papers, etc., and by interviewing cultural experts. Among other potential applications, we identified the evaluation of CQ for application domains covered in our system.

Ang et al. [16] developed a self-assessment questionnaire which has 20 items that measure CQ. This questionnaire was validated across samples (n=1564), time, countries and method of measurements. This questionnaire was used to collect data for studies on the test subjects regarding their capacity for cultural adaptation. The questionnaire is generally divided into four sections: metacognition, cognition, motivation and behavior. For example, one of the items is: "*I am conscious of the cultural knowledge I use when I interact with people with different cultural backgrounds*." Van Dyne et al. [17] developed a version of the questionnaire from the point of view of an observer. It is also based on the 20 items of Ang et al. [16] in order to measure CQ in individuals. The questionnaire was adapted from each item of the self-assessment questionnaire to reflect the assessment made by an observer rather than the user himself. As explained by Van Dyne et al. [17], these questionnaires allow for the effective assessment of CQ by cultural experts in practical applications. It is difficult to evaluate users only by these questionnaires without any cultural experts present. Thus, we adapted the self-assessment questionnaire and the observer questionnaire to measure CQ in order to integrate the CQ experts' knowledge, for the purpose of evaluation and recommendation functions offered by our system. Users can therefore be evaluated, and appropriate suggestions can be offered by the system.

4 Cultural Intelligence Computational Model

When processed by humans through questionnaires, CQ generally has two types of data: the first type is associated with "hard" computing, which uses numbers, or crisp values; the second type is associated with "soft" computing, which operates with uncertain, incomplete and imprecise soft data. The second type is presented in a way that reflects human thinking. When we explain the cultural concept of cross-cultural activities, we usually use soft values represented by words rather than by crisp numbers. Traditional techniques, or "hard" computing, cannot treat CQ soft data. In order to enable computers to emulate human-like thinking and to model a human-like understanding of words, we use a hybrid neuro-fuzzy technology to invent a CQ computational model. This soft-computing technology is capable of dealing with uncertain, imprecise and incomplete CQ soft data.

The hybrid neuro-fuzzy technology makes use of the advantages and power of fuzzy logic and the ANN. The hybrid technology represents the essence of our computational model. The CQ computational model is based on the four-dimensional structure of Ang et al. [16]. The model is noteworthy because we clearly put forward and use that four CQ dimensions make up an integrated and interdependent entities. Essentially, the computational model is a multi-layer neural network with the functional equivalency of a fuzzy inference process. This neural network is not a simple neural network due to all of the cultural rules embodied in these structure nodes. The neuro-fuzzy network is composed of six layers in our computational model. The model is shown in Fig. 1. This hybrid computational model has 20 inputs which represent the 20 items of the questionnaires to measure CQ: the metacognitive dimension (MC) has four items, the cognitive dimension (C) contains six items, the motivational dimension (M) includes five items and the behavioral dimension (BEH) consists of five items and has one output: CQ.

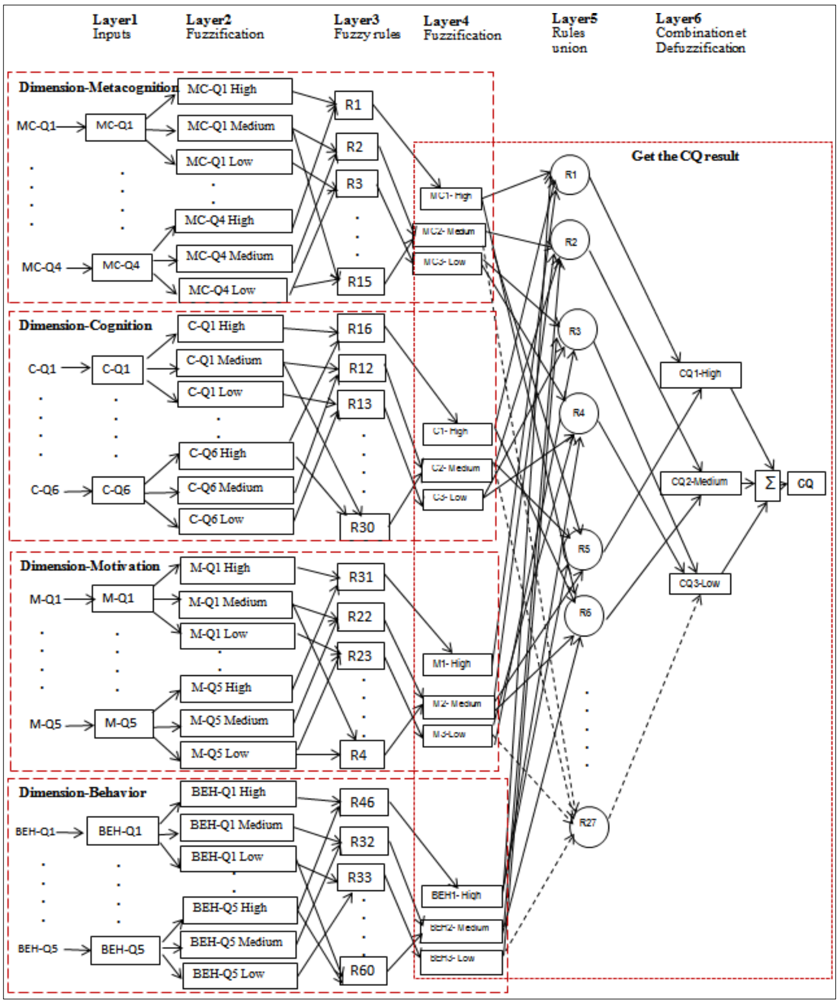


Fig. 1. Computational Model of Cultural Intelligence

Layer 1 - Input: No calculation is made in this layer. Each of the 20 neurons corresponds to an input variable. These input values are transmitted directly to the next layer.

Layer 2 - Fuzzification: Each neuron corresponds to a linguistic label. Fuzzy linguistic variables used in our model are triangular membership functions (e.g., High, Medium and Low), associated with one of the input variables in Layer 1. We have 60 neurons in this layer.

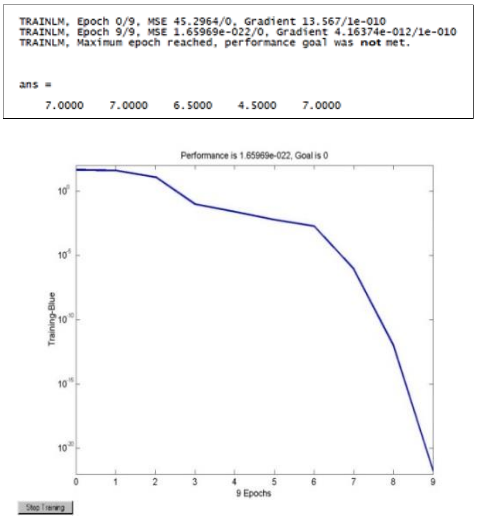
Layer 3 - Fuzzy Rules: The output of a neuron at this layer is the fuzzy rules of CQ. For example, Neuron R1 represents Rule 1 and receives input from the neurons MC-Q1 (High) and MC-Q4 (High), etc.

Layer 4 – Fuzzification: In this layer, the neurons receive the membership degrees as the inputs which are produced from the fuzzy rules layer.

Layer 5 - Rule Unions (or consequence): This layer has two main tasks: 1) to combine the new precedent of rules; and 2) to determine the output level (High, Medium and Low) which belongs to the CQ linguistic variables. For example, R1 is the input of MC1 (High) and C1 (High), etc. It integrates the four dimensions of CQ to make a logical judgment in this layer by using 27 CQ rules.

Layer 6 - Combination and Defuzzification: This layer combines all the consequence rules and, lastly, computes the crisp output after Defuzzification. This layer has three neurons: CQ-High, CQ-Medium and CQ-Low. The Center of Gravity method is used to calculate the output.

This multilayer neuro-fuzzy network can apply standard learning algorithms (such as back-propagation) to train it. This mechanism is very useful, especially in those situations where cultural experts are unable to verbalize which knowledge or problem-solving strategy they use. To illustrate how the computational model learns, consider an example from this model shown in Fig. 2.



```
TRAINLM, Epoch 0/9, MSE 45.2964/0, Gradient 13.567/1e-010
TRAINLM, Epoch 9/9, MSE 1.65969e-022/0, Gradient 4.16374e-012/1e-010
TRAINLM, Maximum epoch reached, performance goal was not met.


ans =

    7.0000    7.0000    6.5000    4.5000    7.0000
```

Fig. 2. Learning in the Computational Model

Suppose we have collected five people's answers as input data, and get five corresponding CQ evaluation results from the output of the model as: *y* = [5, 6, 7, 3, 2]. For any reason, the cultural experts gave five desired CQ output values as: *yd* = [7, 7, 6.5, 4.5, 7]. We then used these five pairs of input data and the desired values to train the model. After nine epoch training processes, our new output from the model was: *y* = [7, 7, 6.5, 4.5, 7]. The model's output quite accurately corresponds to the CQ values provided by the cultural experts. In the future, the system should be trained with big data and calibrated consequently.

5 Implementing the Model in an Intelligent System

We would like the system, first, to be capable of acquiring, extracting and analyzing the new CQ knowledge of experts, and second, to serve as an efficient team comprised of top CQ experts, able to provide both recommendations and explanations to users whenever required in culturally diverse settings. Hence, we implemented the computational model in an expert system, called CQES (Cultural Intelligence Evaluation System). Fig. 3 shows the structure of the CQES.

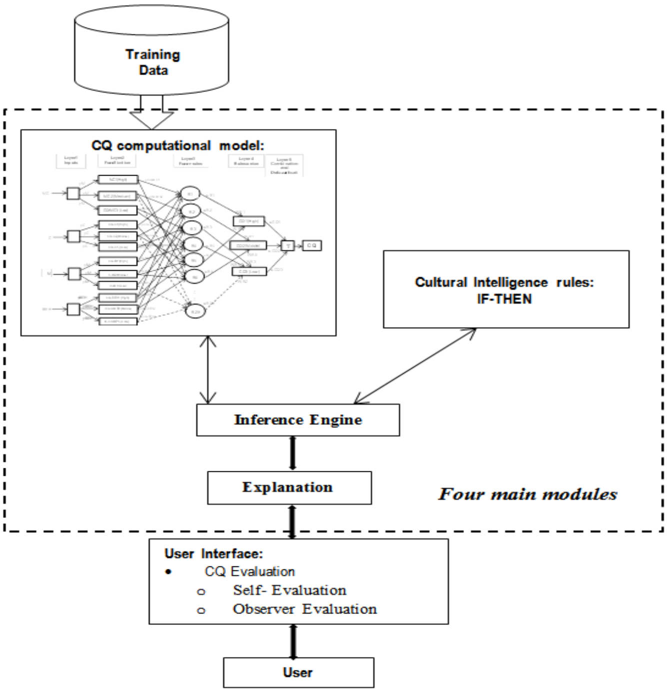


Fig. 3. Structure of CQES

The CQES structure includes four main modules: 1) *The CQ Computational*

Model contains CQ knowledge that is useful for solving CQ problems. The soft-computing technology used in this model enables the system to reason and learn in an uncertain and imprecise CQ setting. It supports all the evaluation steps in the system. This module connects with the *Training Data Database*. The *Training Data Database* are sets of training examples used for training the neuro-fuzzy network during the learning phase. 2) *The Cultural Intelligence Rules* examine the CQ knowledge base, which is represented by the trained network, and produce rules which are implicitly built into and incorporated in the network. 3) *The Inference Engine* controls the flow of information in the system and initiates inference reasoning from the computational model. It also concludes when the system has reached a solution. 4) *The Explanation* explains to the user why and how the CQES reached the specific CQ evaluation results. These explanations include the conclusion, advice and other facts required for deep reasoning. Therefore, the following details explain how users can get two evaluations (self and observer evaluations) using the 20-item questionnaires (see the interface of the CQES in Fig. 4).

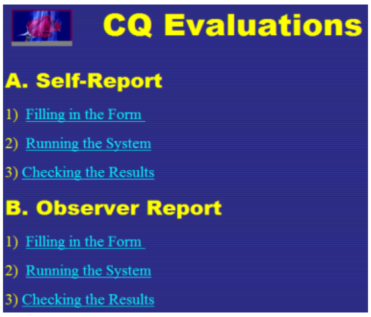
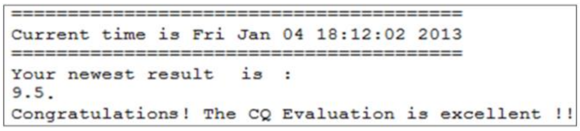


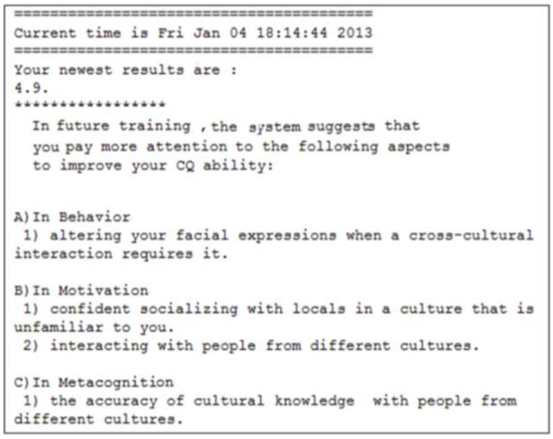
Fig. 4. Interface of CQES

For example, two different results of the self-evaluation questionnaire that evaluate the user's CQ are presented in the CQES as follows:
Result 1: After inputting the answers to the 20 items in the CQES, the system provides the feedback. If a user's evaluation achieves a high score (e.g.: more than 8), the system shows the following message:



```
========================================
Current time is Fri Jan 04 18:12:02 2013
========================================
Your newest result  is  :
9.5.
Congratulations! The CQ Evaluation is excellent !!
```

Result 2: When the evaluation results are below 6, the system accordingly gives useful suggestions for personal self-development as required. This process permits the system to evaluate users so as to identify their problems in the CQ domain and then offers several precise recommendations to users based on the results of the evaluation. Moreover, the system uses natural language to give users recommendations in order to provide them with a stress-free and friendly evaluation. The CQES presents some recommendations as follows:



```
=======================================
Current time is Fri Jan 04 18:14:44 2013
=======================================
Your newest results are :
4.9.
****************
  In future training ,the system suggests that
  you pay more attention to the following aspects
  to improve your CQ ability:


A)In Behavior
 1) altering your facial expressions when a cross-cultural
interaction requires it.

B)In Motivation
 1) confident socializing with locals in a culture that is
unfamiliar to you.
 2) interacting with people from different cultures.

C)In Metacognition
 1) the accuracy of cultural knowledge  with people from
different cultures.
```

Organizations could also use the CQES (both self- and observer evaluations) to evaluate and train employees so that the latter may function more effectively in such situations. We envisage that CQES could effectively be integrated in a CATS to offer training in culture intelligence based on the assessment provided by CQES.

6 Conclusion

This research is original and attempts to give a productive solution by replacing or supporting CQ experts with computers for assessing and provide recommendations for training. This innovative research has managed to computerize the underlying principles of CQ in order to help individuals to improve their ability to adapt to a new culture.

The main contributions of this research are: inventing a CQ computational model and implementing the model in an expert system called CQES. As a ‘culturally aware’ intelligent system, the CQES can be used to train individuals in CQ training by providing them with evaluation, and specific suggestions to improve their weaknesses in the corresponding area. This point is of particular importance in modern learning theories.

7 References

1. Blanchard, E. G., Mizoguchi, R., Lajoie, S. P. (2010). Structuring the cultural domain with an upper ontology of culture. Handbook of research on culturally aware information technology: Perspectives and models, IGI Global, Hershey PA.

2. Lane, H. C., Hays, M. J (2008). Getting down to business: Teaching cross-cultural social interaction skills in a serious game. Culturally-Aware Tutoring Systems, ITS 2008, Montreal, Canada, June 23-27

3. Lewis Johnson, W., Rickel, J. W., Lester, J. C. (2000). Animated pedagogical agents: Face-to-face interaction in interactive learning environments. International Journal of Artificial intelligence in education 11.1. pp.47-78

4. Earley, P.C., Ang, S (2003). Cultural intelligence: Individual interactions across cultures. Stanford, CA: Stanford University Press.

5. Wu, Z. X., Nkambou, R., Bourdeau, J. (2012). The Application of AI to Cultural Intelligence. The 2012 World Congress in Computer Science, Computer Engineering and Applied Computing, ICAI 2012, The International Conference on Artificial Intelligence, Las Vegas, U.S.A.

6. Hofstede, G. (1980). Culture's consequences: International differences in work-related values. London: Sage Publications.

7. Sperber, D. (1996). Explaining culture: A naturalistic approach, Oxford: Blackwell.

8. Kroeber, A. L., Kluckhohn, C., Untereiner, W., Meyer, A. G. (1952). Culture: A critical review of concepts and definitions (Vol. 47, No. 1). New York: Vintage Books.

9. Bennett, J. M., Bennett, M. J. (2004). Developing Inter-cultural Sensitivity: An Integrative Approach to Global and Domestic Diversity. In Dan Landis, Janet M. Bennett, & Milton J. Bennett (Eds.), Handbook of Intercultural Training, 3rd ed., pp. 147–165

10. Earley, P. C., Mosakowski, E. (2004). Cultural Intelligence. Harvard Business Review, 82, pp.139–146

11. Peterson, B. (2004). Cultural intelligence: A guide to working with people from other cultures. Yarmouth, ME: Intercultural Press.

12. Brisling, R., Worthley, R., MacNab. (2006). Cultural Intelligence: understanding behaviors that serve people's goals. Group and organization management.

13. Thomas, D. C., Inkson, K. (2005). Cultural Intelligence People Skills for a Global Workforce. Consulting to Management, vol. 16 (1). March. pp. 5-9

14. Johnson, J. P., Lenartowicz, T., Apud, S. (2006) Cross-cultural Competence in International Business: Toward a Definition and a Model. Journal of International Business Studies, vol. 37. pp. 525–543

15. Ang, S., Van Dyne, L. (2008). Conceptualization of Cultural Intelligence. Handbook on cultural intelligence: Theory, measurement and applications, Chapter I, pp.1-15. Armonk, NY: M.E. Sharpe.

16. Ang, S., Van Dyne, L. (2010). Handbook of Cultural Intelligence. 1st ed. M.E. Sharpe. Armonk.

17. Van Dyne, L., Ang, S., Koh, C. (2008). Development and Validation of the CQS: The cultural intelligence scale. Handbook of Cultural Intelligence. 1st ed. M.E. Sharpe, Armonk.

Cross-Cultural Differences and Learning Technologies for the Developing World

Workshop Co-Chairs:

Ivon Arroyo
Worcester Polytechnic Institute, Worcester, MA, United States

Imran Zualkernan
American University of Sharjah, United Arab Emirates

Beverly P. Woolf
University of Massachusetts Amherst, Amherst, MA, United States

http://cadmium.cs.umass.edu/LT4D

Preface

The Learning Technologies for the developing world (LT4D) workshop aims to provide a forum for a discussion of corss-cultural differences and rational introduction of learning technologies in the developing world, exploring the economic constraints, socio-cultural differences, political and other constraints that shape the implementation and the affordances for learning technologies in the developing world.

Besides differences in socialization and cultural differences, well-intentioned introduction of learning technologies in developing countries can fail for mundane reasons such as teachers not willing to use the technology because of lack of comfort with technology, or simply lack of computers in sharp contrast to abundance of mobile devices.

Such constraints cannot be ignored. Rather than blindly implanting technologies, based on a rationalized discussion of such issue and constraints, and possibilities for the immersion of learning technologies, the workshop then aims to provide future visions and roadmaps of such technologies for the developing world and subsequent practical implementation for technology enhanced learning.

The workshop intends to touch on the following broad set of issues:

1. Cross-cultural differences in educational outcomes of AIED systems or non-adaptive learning technologies across countries, developing vs. developed, or across developing countries.
2. Ideas to solve issues of economic cost of adapting interactive learning environments (ILEs) to developing countries
3. Examples of Localization and Cultural translation of systems and interfaces
4. Issues of Social Inclusion: how to encourage and support both individuals and communities that are marginalized --economically, socially, or culturally; indigenous communities, and other special communities.
5. Science of sustainable design of learning technologies for the developing world; Sustainability of projects in the developing world; funding sources; ideas for maintaining technology resources and personnel
6. How education technology is used in the developing world; how is or should it be used? As a means? As an end?
7. Supporting Teacher Training via e-Learning in developing countries
8. How can Educational Data Mining help to support education and reveal information that would help developing countries
9. Differences in realities and possibilities of implementation of Interactive Learning Environments (ILEs) across the developing world? Is there a common ground, or are countries too different from each other?
10. Issues of timing: Are there key areas where learning technologies can have an immediate impact?
11. Models of adoption (or non-adoption) of learning technologies in the developing world

12. An analysis of great successes or drastic failures in applying ILEs to the developing world
13. Opportunities for leap frogging and avoiding mistakes in the developed world

July, 2013
Ivon Arroyo, Imran Zualkernan, Beverly Woolf

Program Committee

Table of Contents

Towards Localization of Automated Tutors for Developing Countries

Imran Zualkernan[1,], Ivon Arroyo[2], Beverly P. Woolf[3]

[1] American University of Sharjah, United Arab Emirates
[2] Social Sciences and Policy Studies, Worcester Polytechnic Institute
[2] School of Computer Science, University of Massachusetts Amherst

izualkernan@aus.edu, iarroyo@wpi.edu, bev@cs.umass.edu

Abstract. This paper describes localization issues in relation to AIED systems in the developing world, and analyzes the particular case of the successful immersion of learning technologies to schools in Pakistan. The paper analyzes the needs for personalized learning in the developing world in comparison to countries such as the United States. A model and a survey based on various types of localization dimensions like teacher, student, and cultural alignment was developed and deployed to conduct an evaluation of an AI tutor called the Wayang Outpost in Pakistan. The results are that teachers are likely to use such a system if available, and that their intention to use such a tutor in the future is strongly dependent on how well the tutor is aligned with their teaching practices, students' learning habits, and whether the language in the tutor is understood by students. On average these teachers were also willing to allocate about two hours per week for such automated tutors.

Keywords: Developing World, Adaptive Technologies, Localization

1. Introduction

A recent study [1] that used a self-contained traveling van to deliver Khan Academy (www.khanacademy.org) videos in conjunction with Android-based online assessments for children resulted in two interesting observations. First, a learning technology intervention for a seemingly culture-agnostic subject like grade 4 and 5 Mathematics required a significant effort in 'localization' that went far beyond language translation [2]. Second, the statistical effects observed between treatment and control groups were high when compared with those obtained using automated tutors in the West [1],[3]. Taken together, these two observations suggest that while there is a great potential for using automated tutors in developing countries, to be effective, these tutors may need to undergo extensive localization along a number of non-obvious dimensions.

Adaptive tutoring systems have been effective at improving students' achievement in a variety of tests, including standardized tests. For instance, the Wayang Outpost Mathematics Adaptive Tutoring System has shown improvements within 0.3-0.8 effect sizes on standardized tests compared classroom instruction, after controlling for time [4]. The Algebra Tutor has shown effect sizes of 0.3-1.2 standard deviation on a variety of tests [5]. Andes tutor for Physics has shown effect sizes of 0.92 [6]. This is particularly impressive considering that human tutors (subject-matter experts working synchronously with a single student) are one standard deviation better compared to a teacher in front of the classroom with a typical class size of 20 students. This is in contrast to what was originally believed in studies by Bloom [7], which claimed that a human tutor could be 2 standard deviations better than classroom instruction. Since the van study using Khan Academy cited earlier [1] showed effects of 0.87 to over 3, it is expected that the use of adaptive tutors in developing countries will yield much better results. However, as indicated earlier, when considering the implementation of tutoring systems in countries other than the United States, and particularly in the developing world, the important question that urges to be answered is whether and how much *localization* efforts are required to make these adaptive tutors be effective.

The remaining article shows that, while much language and cultural translation needs to be carried out, there is potential to achieve large effect sizes, and that there are specific needs of the developing world that make the use of adaptive learning environments particularly ideal for students with a large variety of unique circumstances. However, cultural differences need to be understood, as they can affect the ecological validity of the intervention, and the general effectiveness of the teaching tool.

2. Why Adaptive Tutors for Developing Countries?

This section motivates the need for automated tutors in developing countries.

2.1 Teachers and Students

Quality and availability of teachers is a key input into the educational quality of children. However, according to a recent study [8], 29 countries mostly in Arab or Sub-Saharan Africa regions have severe teacher gaps and need to grow annually by at least 3.0% during the 2010 to 2015 period. Even when the teachers are present, teacher absenteeism remains a problem, ranging from 3 percent in Malawi to 27 percent in Uganda [9]. [10] reports that teachers absenteeism in six developing countries was about 19%, with Peru at 11%, Indonesia at 19%, and in India at 25%. In a meta-study of developing country learning interventions from 1990 to 2010, out of 79 studies, 5 studies showed mostly negative impacts of teacher absenteeism [11].

There is also a wide variation (800-1000 hours) in the contact time between teacher and students in the developing world [12]. However, more contact time with teachers does increase performance in majority of cases [12]. There are also differences between various countries when it comes what a teacher actually does in the classroom. For instance, in Tunisia, Morocco, Brazil and Ghana, the students were engaged in learning 79%, 71%, 63% and 39% of the time respectively [13]. However, teachers in these countries mostly used “chalk-and-talk” which can result in limited student attention and subsequent recall [13]. Automated tutors have a potential to bring standardized learning processes to children and unlike teachers, tutors are less likely to suffer from absenteeism.

Pupil-teacher ratios are also much higher in the developing countries than the West [14]. For example, in US and UK, teacher-pupil ratios are 14 and 18 respectively. These ratios can be as high as 65 (Rwanda) or Zambia (68) in sub-Saharan Africa. South Asian countries like Pakistan and Bangladesh have pupil-teacher ratios of 40 and 43 respectively. However, a better pupil-teacher ratio does not necessarily guarantee better student performance. For example, [15] observed that reducing the pupil-teacher ratio alone from 88 to 40 did not have a significant impact on learning. However, contract teachers and a strong institutional PTA support did have a positive impact. A meta-study in developing countries also showed that out of 101 studies, 59 studies showed a negative impact of the larger class size, but only 30 studies were statistically significant [12]. Surprisingly, however, another 39 studies showed a positive effect, out of which 15 were statistically significant. Another meta-study of developing countries also showed that effect of teacher-pupil ratio alone on student performance was inconclusive [11].

In summary, in developing countries, there is a shortage of teachers, high teacher absenteeism, and use of ineffective pedagogical approaches, and high pupil-teacher ratios which makes automated tutors a good choice.

2.2 Alternatives to Tutors: Better Textbooks and More Homework assignments

One argument could be that rather than supplying schools with adaptive tutors that require computers and other additional infrastructure, perhaps better teaching materials is all that is required. However, providing textbooks to children in Kenya did not raise test scores of students overall, but did increase the scores for higher performing students suggesting that these books were primarily targeted to the smarter students [11]. A Meta-study of developing countries also confirmed that there is little evidence that just providing textbooks, workbooks and exercise books increased student learning [12].

While there is some debate about whether more homework impacts student performance, there is a general correlation between quality and quantity of homework and student achievement in the West (e.g., [16]). This trend also seems to hold for developing countries. For example, in a meta-study of developing country interventions from 1990 to 2010, 5 studies showed mostly positive impact of more homework [11]. However, one key problem is that in impoverished regions of many developing countries, children have to work after school leaving little time to do homework. For example, [17] observed that in a survey of 1030 children in three parts of South Africa, 26.5% of the children working on farms missed school, arrived late or were too tired to do their homework. Similarly, [13] points out that in a country like Ghana, 84% of the parents and 54% of children reported spending less than one hour per day on their homework. In contrast, technology is used in the United States for students to do homework, and provide immediate assessments to the teacher who can guide discussions about the questions that were wrong [18]. Merely showing that a question is wrong, without any further hints or explanations, provided a 0.5 standard deviation of improvement compared to not getting any immediate marking that a homework question was wrong.

2.3 What is known about Personalized Instruction and Adaptive Tutors?

This section summarizes what is known about adaptive learning and automated tutors (not necessarily adaptive) in developing countries and types of effects achieved by doing so. A large study of 140 schools in Keyna shows that the simple act of splitting students into two sections based on ability did have an impact, and effects of 0.14 to 0.18 were achieved [15]. Similarly, using specialized human tutors for remedial classes yielded performance effects of 0.18 to 0.28, while use of an automated tutor achieved 0.35 to 0.47 effects in Mathematics [19]. These two studies show that personalizing instruction at the group level works, and clearly automated tutors also tend to have an impact.

Another related issue is whether the tutor should complement, or replace existing instruction in the classroom. For example, [20] found that a complementary program where children were actually provided computer-aided learning had an effect of 0.28 as opposed to the replacement group whose performance actually got worse (-0.57); the replacement group replaced conventional teaching with an automated tutor. This tends to suggest that automated tutors can perhaps be more effective in a complementary mode.

2.4 Base Grades are lower and Variability is high

The benefits of personalization should be high especially if there is large variability in student achievement within the same classroom, so that the instruction by one teacher might not fit the needs of every student. We have data from a series of studies in the United States, which show performance in a standardized test for both high achieving and low achieving schools, urban vs. rural, in standardized tests that should have good psychometric properties. The test is the Northwest Evaluation Association MAP test, which evaluates students' Mathematics expertise across grades and at different points of the year. Table 1 shows scores of students in two schools during the 2012 academic year.

It can be seen from Table 1 that the standard deviations are small for the NWEA MAP test, which is a test administered by many schools across the United States. The standard deviations for the rural-area high achieving school are 15% of the full range of scores recorded by 7-8 grade students (note: *stdev / (maximum score – minimum score)* = 14.6/(267-167) = 0.146), and 20% of the full range of scores recorded by the low achieving urban school, for grades 9-10 (16.8/(262-180)).

Table 1. Variation in NWEA's MAP standardized test, in two schools in the United States

School	Grade	N	Median	Mean	Natl. Avg.	Std. Dev.	Min.	Max.	Range
Urban-Area School	9-10	97	221	220	234	16.8	180	262	82
Rural-Area School	7-8	223	234	233	228	14.6	167	267	100
Total						16.3	167	267	

No standardized testing data are available for Pakistan. However, results of a standardized Mathematics tests for grade 4 and 5 in semi-rural schools for twenty different school sections (18 different schools) from a recent study [1] show that this ratio (SD/Range) is higher than 20% (Mean = 26.71; SD = 2.97; minimum=21; maximum = 30; Anderson-Darlington, $p>0.05$; single sample t-test; DF = 19, T=10.09, $p<0.05$). In other words, in these semi-rural schools of Pakistan, the standard deviation is more than 20% of the range and sometimes as high as 30% showing more diversity in the learning achievements of students than their counterparts in the United States. Because of this higher variability, it is expected that automated and adaptive tutors will be more effective in developing countries like Pakistan.

Another key issue for students in developing countries is that the overall competencies tend to be much lower than students in the West. For example, in India only 19.5% of third grade children in Vadodara, and 33.7% in Mumbai, passed the grade one competencies (number recognition, counting and one digit addition and subtraction) in Mathematics [19]. Table 2 shows the Mathematics results from the *Programme for International Student Assessment* (PISA) standardized exam for a few third-world countries; Pakistan does not participate in PISA. As can be clearly seen, large proportions of children in a variety of developing countries from different continents tend to have lower proficiency in Mathematics skills than students from Western countries like the United States.

Table 2. Students in Developing Countries are at lowest proficiency of PISA Mathematics Scores

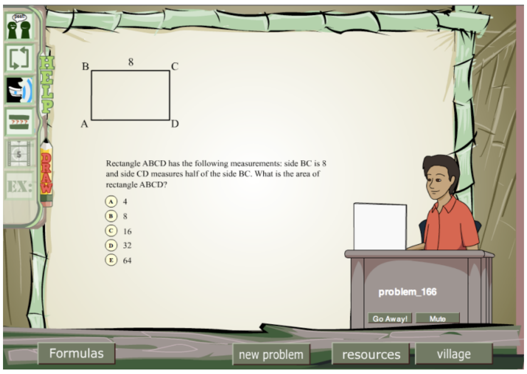
Country	% Students in the lowest proficiency
United States	8
Kazakhstan	30
Trinidad	30
Jordan	35
Argentina	37
Brazil	38
Columbia	39
Albania	40
Tunisia	43
Indonesia	44
Peru	48
Panama	51
Krygyz Republic	65

3 Case Study

As a first step towards localization, a case study was conducted to evaluate how teachers in Pakistan would respond to the use of an interactive adaptive tutor for their students.

3.1 Wayang Outpost – The Adaptive Tutor

An intelligent mathematics tutor for grades 5-12 developed at UMass-Amherst and named Wayang Outpost [4] was selected for this study (See Figure 1). Wayang Outpost has been used by thousands of students in the United States, and students using Wayang Outpost have consistently shown significant learning gains since 2003 on Mathematics tests involving standardized tests items (an increase of 20% achievement level after 3 time periods only), and significant gains on state-standard exams compared to control groups (0.5-0.7 standard deviations depending on the study). Students using Wayang Outpost have also improved more on specific areas of a national standardized test compared to control groups (MAP, a national test of NWEA) for specific mathematics knowledge units that were tutored by Wayang Outpost, and not for other areas of mathematics that were not tutored by Wayang Outpost during those sessions.



Rectangle ABCD has the following measurements: side BC is 8 and side CD measures half of the side BC. What is the area of rectangle ABCD?

- (A) 4
- (B) 8
- (C) 16
- (D) 32
- (E) 64

Fig. 1. The Wayang Outpost Math Tutor interface.

Wayang Output targets the United States Mathematics curriculum of grades 6 through 11 covering a large range of topics including number sense, pre-algebra, algebra, geometry, logical reasoning. The pedagogical approach of the Wayang Tutor is based on cognitive apprenticeship and mastery learning, and its internal mechanism of adaptive behavior of item selection is based on empirical estimates of problem difficulty and a variety of parameters that regulate its behavior, which are set by a combination of input from teachers and the researchers [21]. For feedback and scaffolding, Wayang Outpost relies on the Theory of Multimedia Learning, and implements many of its principles, and provides also videos and worked-out examples as part of its support. However, the main mechanism of support consists of hints that solve a small part of the solution for the student, and allow him/her to continue on his own, or ask for more support. Wayang Outpost carries out several instructional tasks: it models (introduces the topic via worked-out examples, making steps explicit, and working through a problem aloud); provides practice with coaching (offering multimedia feedback and hints to sculpt performance to that of an expert's); scaffolds (putting into place strategies and methods to support student learning); provides affective support (via affective characters that reflect about emotions, encourage students to persevere and demystifies misconceptions about mathematics problem solving), and encourages reflection (self-referenced progress charts that allow students to look back and analyze their performance) at key moments of loss, boredom, or un-excitement.

3.2 Survey Design and Data Collection

Based on experiences gained in localizing Khan Academy videos [2], a survey to isolate the various factors that could have an impact on teachers' utilization of Wayang Post was designed as shown in Table 3. The survey was delivered to nine Mathematics teachers on April 28, 2013 in a private urban school in Peshawar, Pakistan. The teachers were introduced to Wayang Outpost, and were led through a one hour session as a student through Wayang Outpost. Each teacher then filled out a survey shown earlier where each item was scored on a Licker-type scale with 1 = Strongly Agree and 5=Strongly Disagree.

3.3 Results

The teachers thought that they could spare on average of about 2 hours per week for an automated tutor session (Mean = 2.44; SD=1.13). This is a substantial amount of time considering that the teachers in this school spend a total of about 5 hours per week on teaching Mathematics. The statistics for the remaining factors are shown in Table 4.

Authenticity (A) and Cultural Alignment (CA) were dropped from further analysis because the value of Cronbach's alpha were lower than 0.7 indicating a lack of internal consistency in how teachers answered the various items; Cronbach's alpha was higher than 0.7 for all other factors. While BI, LC, PA and TA were normally distributed (Anderson-Darlington; p>0.05), since the total number of respondents was small (n=9), non-parametric analysis was used to analyze the data.

As Table 5 shows, all the internally consistent factors were highly correlated. One key variable in the experiment was whether teachers would use such a tutor in the future (BI). BI can be considered a response variable and based on Ordinal Logistic Regression, BI is strongly affected by PA (G= 13.278, DF = 1, P-Value = 0.000) with an odds ratio of 0.01. BI is also affected by TA in a similar fashion (G = 10.899, DF =

1, P-Value = 0.001) with an odds ratio of 0.02. Finally, BI is also affected by BI (G = 5.678, DF = 1, P-Value = 0.017) but the odds ratio is 0.14 indicating that its impact is lesser than those of the two other variables.

Table 3. Survey Design to Determine Factors of Teachers's Adoption

Factors	**Items**
Teacher Alignment – How well does the tutor fit with teaching style of the teacher?	Wayang Outpost System fits well with the way I teach Mathematics
	Wayang Outpost is consistent with how I like my students to learn Mathametics
	Wayang Outpost teaches Mathematics the way I teach it
Language Comprehension – How well does the child comprehend the language used in the tutor?	The children will understand the language used in Wayang Outpost
	The children will not have any difficulty reading the problems posed in Wayang Outpost
	The children will find it easy to follow the problem descriptions and feedback in Wayang Outpost
Authenticity – How authentic are the problem being posed in the tutor?	The children can relate to the problems being posed in Wayang Outpost
	The examples in Wayang Outpost are consistent with how these children live their lives
	The problems in Wayang Outpost are about things that these children care about
Cultural Alignment – Is the tutor aligned with cultural norms and taboos?	The problems in Wayang Outpost do not violate any traditions or taboos
	The problems in Wayang Outpost are consistent with the Pakistani culture
	The problems in Wayang Outpost are not alien to children from a cultural perspective
Pedagogical Alignment – Is the tutor consistent with how children learn?	Children will not have any difficulty following the way Wayang Outpost teaches
	Children will enjoy interacting with the various characters that help them out while solving problems using Wayang Outpost
	Children will like solving problems and getting feedback on their performance using Wayang Outpost
Behavioral Intention – What is the likely-hood of the teacher using the tutor in the near future?	If Wayang Outpost were available, I would use it in my classroom
	I would like to use Wayang Outpost for teaching Mathematics
	It would be great to use a system similar to Wayang Outpost to teach Mathematics

Table 4. Summary of Survey Responses (n=9)

Factor	**Mean**	**StDev**	**Median**	**Min.**	**Max.**	**Median [95% Conf. Intrval]**
Authenticity (A)	2.70	0.61	2.66	1.66	3.33	2.67 [2.17, 3.33]
Behavioral Intention (BI)	**2.29**	**0.82**	**2**	**1**	**3.66**	**2.17 [1.67, 3:00]**
Cultural Alignment (CA)	3.18	0.62	3.33	2.33	4	3.17 [2.67, 3.67]
Language Comprehension (LC)	2.96	0.94	3	1.33	4	3 [2.17, 3.67]

Pedagogical Alignment (PA)	2.25	0.83	2.33	1	3.33	2.33 [1.50, 3.67]
Teacher Alignment (TA)	2.29	0.77	2.333	1	3.33	2.33 [1.67, 3:00]

Table 5. Correlation between the Various Factors (Kendall-Tau; * = p<0.05)

	LC	**BI**	**TA**	**PA**	**CA**
BI	0.627*	1			
TA	0.618*	0.746*	1		
PA	0.618*	0.806*	0.941*	1	
CA	0.462	0.344	0.4	0.462	1
A	0.576*	0.646*	0.667*	0.637*	0.381

In summary, the data show that there was a reasonable probability that the teachers would use the system if available, and were willing to allocate about two hours per week for this activity. Further, their intention to use Wayang Outpost of or a similar system is contingent on teacher and pedagogical alignment, and whether Wayang Outpost's language would be understood by the children. However, is important to note that as Table 4 shows, while teachers were not negative about any of the factors, they were mostly not sure (closer to Neither Agree nor Disagree) about pedagogical and teacher alignment etc. This strongly implies the need to consider using these factors in localization of Wayang Post.

5 Conclusion and Future Work

While adaptive tutors like the Wayang Post have shown considerable impact in improving learning outcomes in countries like the United States, an exploitation of the full potential of such systems in the developing world is contingent on careful localization that goes beyond simple language translation. Clearly, there is a dire need for such systems in the developing world and even though the sample size was small, this paper shows that teachers in a developing country are likely to adopt such systems provided the issues of teacher, student alignment etc. are adequately addressed. The challenge now remains to find the resources to localize and deploy adaptive tutors such as Wayang Outpost.

Acknowledgements

This research was funded by an award from the National Science Foundation, NSF # 1109642, REESE, Personalized Learning: strategies to respond to distress and promote success, Ivon Arroyo (PI), with Beverly Woolf and Winslow Burleson. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies. The authors would also like to thank TeleTaleem (pvt) Limited, Islamabad, Pakistan for help with the logistics of conducting the teacher study.

References

1. Zualkernan, I. A. & Karim A. (2013a). Using a Traveling Van to deliver Blended Learning in a Developing Country, ICALT 2013, Beijing, July, 2013 (to appear).
2. Zualkernan, I. A. & Karim, A. (2013b). Zualkernan, I. A. & Karim A. Online Content Localization for Blended Learning in Developing Countries: A Case Study Using Khan's Academy, Proceedings of the 7th International Technology, Education and Development Conference, Valencia (Spain), March 4-6, 2013.
3. VanLehn, H. K. (2011). The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems, Educational Psychologist, vol. 46, no. 4, 2011, pp.197-221.
4. Arroyo, I,, Beal, C.R,, Murray, T,, Walles, R, Woolf, B.P. (2004) Web-Based Intelligent Multimedia Tutoring for High Stakes Achievement Tests. Proceedings of 7th International Conference on Intelligent Tutoring Systems Conference. Maceio, Brazil. LNCS, Springer.
5. Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. (1997). Intelligent tutoring goes to school in the big city. International Journal of Artificial Intelligence in Education, 8, 30-43
6. VanLehn, K., Lynch, C., Schulze, K. Shapiro, J. A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., & Wintersgill, M. (2005). The Andes physics tutoring system: Five years of evaluations. In G. McCalla, C. K. Looi, B. Bredeweg & J. Breuker (Eds.), Artificial Intelligence in Education. (pp. 678-685) Amsterdam, Netherlands: IOS Press.
7. Bloom, B. S. (1984). The 2 sigma problem: The search formethods of group instruction as effective as one-to-one tutoring. Educational Researcher
8. CTTG (2012). Closing the Trained Teacher Gap, Global Campaign for Education, Rosebank, Johannesburg, South Africa, 2012.
9. Guerrero, G., Leon, J., Zapata, M., Sugimaru, C. & Cueto, S. (2012). What works to improve teacher attendance in developing countries? A systematic review, London: EPPICentre, Social Science Research Unit, Institute of Education, University of London.
10. Chaudhury, N., Hammer, J., Kremer, M., Muralidharan, K. & Rogers, F. S. (2006). Journal of Economic Perspective, Volume 20, Number 1, Winter 2006, Pages 91–116.
11. Glewwe, P. W., Hanushek, E. A., Humpage, S. D., Ravina, R., (2010). School Resources and Educational Outcomes in Developing Countries: A Review of the Literature from 1990 to 2010, Working Paper 17554, Available at: http://www.nber.org/papers/w17554.
12. USAID (2012). USAID Education Strategy Reference Materials, USAID, April, 2012.
13. Abadzi, H. (2007). Absenteeism and Beyond: Instructional Time Loss and Consequences, The World Bank Independent Evaluation Group, Thematic, and Global Evaluation Division, October 2007.
14. WBI. (2012). World Bank Development Indicators, The World Bank, Washington, D.C.
15. Duflo,E., Dupas,P. & Kremer, M. (2011). Peer Effects, Teacher Incentives, and the Impact of Tracking: Evidence from a Randomized Evaluation in Kenya, *American Economic Review 101* (August 2011): 1739–1774.
16. CCLC. (2009). A systematic review of literature examining the impact of homework on academic achievement, Canadian Council on Learning, Canada.
17. Dewes, A. (2012). Agricultural work in South Africa: A contested Space. in *Childhood Poverty: Multidisciplinary Approaches* edited by Jo Boyden, Michael Bourdillon, 2012.
18. Kelly, Y., Heffernan, N., Heffernan, C., Goldman, S., Pellegrino, G. & Soffer, D. (submitted). Estimating the Effect of Web-Based Homework. Proceedings of the 16th International Conference on Artificial Intelligence in Education. Memphis, TN. 2013
19. Banerjee, A. V., Cole, S., Duflo, E., & Linden, L. (2007). Remedying Education: Evidence from Two Randomized Experiments in India, The Quarterly Journal of Economics (2007) 122(3): 1235-1264.
20. Linden, L. (2008). Complement or Substitute? The Effect of Technology on Student Achievement in India, MIT Jameel Poverty Action Lab, Cambridge, Mass.
21. Arroyo, I., Mehranian, H, Woolf, B.P. (2010) Effort-based Tutoring: An Empirical Approach to Intelligent Tutoring. Proceedings of the Third International Conference on Educational Data Mining. Pittsburgh, PA.

A Case Study of the Localization of an Intelligent Tutoring System

Phaedra Mohammed and Permanand Mohan

Department of Computing and Information Technology, The University of the West Indies,
St. Augustine, Trinidad and Tobago
{phaedra.mohammed@gmail.com, pmohan@tstt.net.tt}

Abstract. The matter of a one-size-fits-all approach towards the development of culturally-relevant educational software is debated with one side arguing for internationalization and the other side arguing for localization. This paper takes a pragmatic look at the issues involved in localization and aims to shed light on the strengths and limitations of undertaking culture as a design feature. With an emphasis on the application layer, the paper investigates the requirements and steps that need to be taken when using cultural contexts in educational software. It describes the design of a localized intelligent tutoring system developed for the context of Trinidad and Tobago and discusses how the prototype was evaluated in two separate studies which looked at learning gains, students' opinions, attitudes, and preferences for localization.

Keywords: Localization, cultural translation, intelligent tutoring systems

1. Introduction

This paper is set in the context of Trinidad and Tobago. With a GDP of $20,400US [2] Trinidad and Tobago is one of the wealthiest nations in the Caribbean. Liquefied natural gas, petroleum and its byproducts make up the bulk of the country's exports and account for approximately 40% of the country's GDP [7]. The average population size is 1.2 million, life expectancy is estimated at around 72 years, and literacy rates are over 98% for ages 15 and older [2]. The country is becoming more modernized as evidenced by the increasing number of Internet users (growth from 8% in 2000 to 48% of the population in 2012) and large number of cell phone users (over 1.8 million) [2]. Although access to personal computers is not as widespread with roughly less than 20% of the population having access, the government of the country provided free laptops to entry-level students in secondary schools in 2010. The challenge that now arises for the country's education sector is whether the software on these machines can support learning in the context of Trinidad and Tobago.

Accommodating for learner diversity based on cultural backgrounds is becoming a major personalisation focus with the increasing drive towards globalization as evidenced in Trinidad with the distribution of laptops. Despite this drive, the knowledge and processes for incorporating culture have not been clearly defined with automation in mind. The matter of a one-size-fits-all approach towards the development of educational software is debated with one side arguing for internationalization and the other side arguing for localization. Proponents of localized approaches argue that culture increases the credibility, realism, familiarity and acceptance of educational systems with the end result being a higher quality learning experience [4, 5]. On the other hand, techniques have been described for creating internationalized designs that do not target a particular culture and avoid cultural specificity but still cater for the needs of a learner. This viewpoint is based on arguments that cultural designs tend to be cosmetic and stereotypical, suffer from designer bias [1], and are overall difficult to automate [8].

The purpose of this paper is not to take a particular side or argue for or against a particular approach. Rather, this paper takes a pragmatic look at the issues involved, and aims to shed light on the strengths and limitations of undertaking culture as a design feature in educational systems. It focuses on the requirements and steps involved in carrying out localization at the application layer. The paper then investigates the requirements and steps that need to be taken in order to use the cultural context of student in educational software and then describes the design of a localized intelligent tutoring system (ITS) developed for the context of Trinidad and Tobago with these requirements in mind. Next, the paper discusses how the system was evaluated in two separate studies which looked at learning gains, students' opinions and attitudes towards the system, and student preferences for localization. The paper concludes with an outline of the lessons learned from these studies and potential research directions for localized systems.

2. The Strengths and Challenges of Localization

Educational frameworks and systems from different sources are typically repurposed in order to reduce costs associated with content development, to replicate proven results with learning gains, and to set standards in educational curricula. Repurposing of such environments entails some form of localization since the design of user interfaces, the selection of teaching strategies, the format and content of the educational material all vary depending on the contextual background of the developers. In order to relate to students and in some cases not offend others, localization of educational environments has been recommended and there have been benefits cited in the literature for students such as increased motivation levels [3, 6].

There are several challenges associated with localization such that actual systems are limited in practice. Many developers have shied away due to the complexity in reliably representing aspects of a particular culture and because of the ill-defined nature of culture [1, 8]. The costs can be higher in the long run because of the amount of effort required in cataloguing cultural knowledge. In addition, localization requires many pieces of metadata for adaptation to go beyond keyword insertion and colour changes. Learning gains seen in one country are not guaranteed in another since many variables exist across countries such as the system may not be used in the same way as in the original country, there might be limited internet connectivity, different hardware, untrained teachers, or even a different curriculum. Lastly, a localized solution runs the same risk of being as offensive or irrelevant as the original version since a student does not belong to a discrete cultural group, and more than just cultural knowledge is required in order to relate to students; a sound understanding of environmental context is critical.

3. Requirements for Effective Localization at the Application Layer

There are three entry points for localization in educational software systems: the presentation layer, the application layer, and the data layer. The application layer selects, modifies, aggregates, and formats raw content based on the user's input, history of events and intended instructional goals. Firstly, data is needed about the student's country of residence (target country) because this data defines the environmental context that the student is familiar with. Such data is typically available from the country's national statistical office or department and can be used to set the scope of the localization. For example, statistics on population density, economic activity, and religious group distribution can be used. Secondly, demographic data from the student is required in order to model his/her contextual background and to know what type of localization to carry out and to what extent. This data comes from users directly entering information about themselves in a form for instance and can include a user's age, gender, native language and residence location for instance. Thirdly, data is required on the contextual groups in the target country specifically what kinds of contextual elements are familiar, appropriate, and relevant to a specific group. This is needed in order to control the direction of the localization process. Next, language rules and localized natural language terms are needed in order to translate textual content and adapt the cultural meaning of the content to suit the student. Semantic markup on these cultural terms is required in order to know how to use the terms in sentences correctly and also to be able to interpret image context in a culturally appropriate manner. Lastly, real-time localization is required in order to adapt content to suit different student preferences.

4. Localized System Design

This section describes the design of a contextually-relevant Intelligent Tutoring System (ITS). A modular design was chosen because of the complexity involved in delivering culturally-relevant instructional experiences. Flexible alteration and improvement of the component features are easily accommodated as a result. Many of the components featured in the design of the localized web-based ITS are based on traditional ITS components but have been modified to include localized data and functionality. A cultural student model, a pedagogical module, cultural heuristics and, a content repository make up the major architectural units of the design.

The cultural student model records the pedagogical events, performance-related student data, suggested hints and instructional guidance given to the student. Cultural background data is assimilated into this model and is made up of demographic data and cultural influences. The pedagogical module consists of instructional rules that constantly access and update the student model in response to input data from the student and events captured from the screen. They control how and when instructional feedback is given, and they manage the selection and transition of learning activities.

A culturally-relevant instructional approach requires that cultural references made by software systems should be applicable to the learning content, familiar to the students, authentically rendered, and integrated into the context of the instructional material [4, 5]. Cultural rules modify the textual portions of instructional content such as question descriptions, scenarios, hints, and instructional feedback produced by systems. The localization of visual portions of these systems, namely the multimedia related to the learning activities, is also handled by these rules. Textual modifications include customizing the language of the instructional feedback, whereas the multimedia localization involves swapping in cultural assets for generalized assets such as images. The scope of the languages used for localization in this paper is restricted to English-based Creole languages. These languages are useful for educational environments because they foster comfortable learning experiences especially in domains that are seen by students as problematic or difficult to cope with [6]. Textual outputs of the localization process are therefore sentences expressed in a local dialect specifically mesolect forms[1], and feature equivalent cultural lexical terms. In order for these rules to properly localize the learning activities and determine appropriate feedback for the student, the learning materials need to be accompanied with metadata descriptions that identify content that can be localized such as hints, question descriptions, and images.

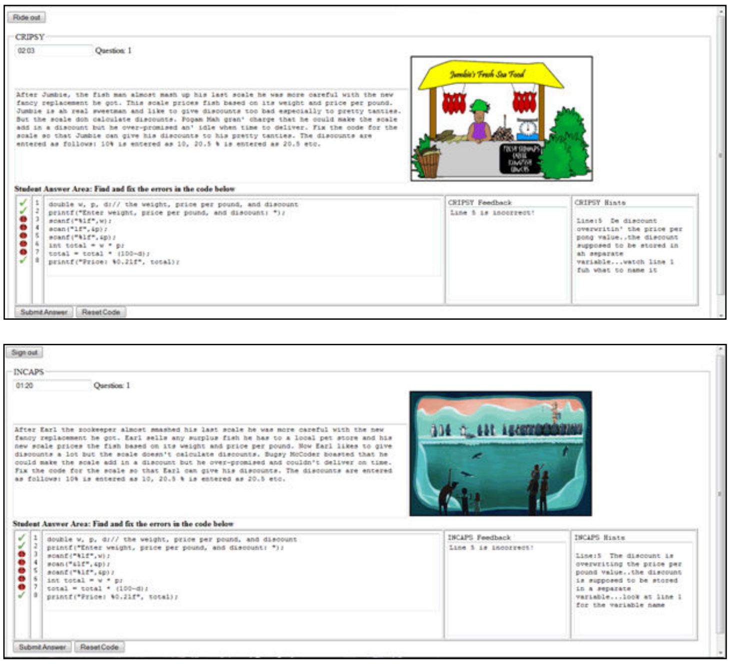
The content repository handles the organisation and distribution of all instructional and cultural assets to a content aggregator. Localized ITSs rely on reusable content more than non-cultural ITSs because of the additional dimension of cultural personalisation; this was the basis for having a separate asset repository - reusability. The content repository primarily hosts all of the educational and interface-related material used by the system and the student model. For example, multimedia files related to the interface's look and feel, such as icons, logos, and those related to the learning exercises (scenario pictures, feedback pictures) are stored here together with educational material such as question descriptions, solutions, feedback files, and topic hierarchies. Each of these assets is described by their asset metadata descriptions which define the context of use and the nature of the assets. These descriptions are indispensable in the design because they facilitate reuse and exchange of compatible assets. Both the pedagogical module and cultural heuristic component use these descriptions when making instructional and localization-related decisions.

5. System Implementation

Two web-based ITSs were implemented based on the software architecture described in the previous section. One system was localized for Trinidad and Tobago's

[1] A variety of language in a Creole continuum that is intermediate between the standard form (acrolect) and forms that diverge greatly from the standard form (basilect).

context (Culturally Relevant Instructional Programming System – CRIPSY) while the other remained generic (Instructional Non-CulturAl Programming System – INCAPS).



After Jumbie, the fish man almost mash up his last scale he was more careful with the new fancy replacement he got. This scale prices fish based on its weight and price per pound. Jumbie is ah real sweetman and like to give discounts too bad especially to pretty tanties. But the scale doh calculate discounts. Pogam Mah gran' charge that he could make the scale add in a discount but he over-promised an' idle when time to deliver. Fix the code for the scale so that Jumbie can give his discounts to his pretty tanties. The discounts are entered as follows: 10% is entered as 10, 20.5 % is entered as 20.5 etc.

Student Answer Area: Find and fix the errors in the code below

```
✓ 1 double w, p, d;// the weight, price per pound, and discount
✓ 2 printf("Enter weight, price per pound, and discount: ");
● 3 scanf("%lf",w);
● 4 scan("lf",&p);
● 5 scanf("%lf",&p);
● 6 int total = w * p;
● 7 total = total * (100-d);
✓ 8 printf("Price: %0.2lf", total);
```

CRIPSY Feedback

Line 5 is incorrect!

CRIPSY Hints

Line:5 De discount overwritin' the price per pong value..the discount supposed to be stored in ah separate variable...watch line 1 fuh what to name it

After Earl the zookeeper almost smashed his last scale he was more careful with the new fancy replacement he got. Earl sells any surplus fish he has to a local pet store and his new scale prices the fish based on its weight and price per pound. Now Earl likes to give discounts a lot but the scale doesn't calculate discounts. Bugsy McCoder boasted that he could make the scale add in a discount but he over-promised and couldn't deliver on time. Fix the code for the scale so that Earl can give his discounts. The discounts are entered as follows: 10% is entered as 10, 20.5 % is entered as 20.5 etc.

Student Answer Area: Find and fix the errors in the code below

```
✓ 1 double w, p, d;// the weight, price per pound, and discount
✓ 2 printf("Enter weight, price per pound, and discount: ");
● 3 scanf("%lf",w);
● 4 scan("%lf",&p);
● 5 scanf("%lf",&p);
● 6 int total = w * p;
● 7 total = total * (100-d);
✓ 8 printf("Price: %0.2lf", total);
```

INCAPS Feedback

Line 5 is incorrect!

INCAPS Hints

Line:5 The discount is overwriting the price per pound value..the discount is supposed to be stored in a separate variable...look at line 1 for the variable name

Fig.1. Screenshots of CRIPSY (top) and INCAPS (bottom) featuring localized and non-cultural versions of the same programming exercise.

Both systems were built for the same educational domain (Computer Science programming) were identical from a functional standpoint, and were implemented using Java-based tools and technology which facilitated seamless integration of the various components into complete systems. The pedagogical module, student model, and cultural heuristic component were implemented using JESS (Java Expert System Shell) rule engines and rules. At the data level, simple file formats were used to manage the content repository since rule engines handle data manipulation primarily using facts. Constructivism and situated cognition were selected as the major instructional strategies since analytical programming skills were being targeted and also because of the good fit between situated cognition and culturally-aware instruction. The development of localized assets for CRIPSY was done semi-automatically and manually. The programming exercise descriptions, parts of the exercise code and instructional hints were localized using subtle, careful use of cultural semiotics, specifically cultural names of objects and foods. As shown in the screenshots in Figure 1 above, both systems used the same instructional content but differed in the expression of the content, that is, cultural and non-cultural. A minimalist interface design was used and instructional feedback consisted of identification of correct/incorrect lines of code, hints for the incorrect lines, and general informative guidance.

6. Prototype Evaluation and Results

A previous study was done using both prototypes and participant details are reported in [6]. That study revealed that the use of both systems resulted in significant increases in student test scores and one design issue affected students in particular: the language used in the localized system. The density of the localisation distorted the system's content into basilect[2] Creole which for some students made the sentences difficult to read quickly and therefore difficult to understand quickly. This observation stimulated the consideration of incorporating a customizable language density scale in the localized system so that students may adjust the language to suit their own preferences. The desirability of such a feature was assessed in the second study.

The second study aimed to find out which system appealed more to the students, what kinds of conditions would impact if at all upon their preferences, and whether a language localization slider would be a desirable feature and why. Fifty-eight (58) students from the previous study participated in this study. The control and test groups were switched so that students would have used both systems at the end of this study. A similar procedure was followed where each student's username activated their newly assigned system and the systems timed out after 30 minutes. A short questionnaire was administered and the results are shown in Table 1 below. After using both systems, more students (68.9%) preferred the localized system (CRIPSY) over the non-localized one (INCAPS). The majority of students reported that they wanted the localization slider (79.3%) while the minority did not see the slider as necessary. Increasing question difficulty did not seem to influence student preference for either system. Lastly, more students (13.8%) changed their preferences from the localized system to the non-localized system when server glitches or software problems occurred compared to those whose preferences were reversed in favor of the localized system (3.4%).

Feedback Topic	Student Preferences	Percentage of Students
Localization Slider	Wanted localization slider	79.3
	Did not want localization slider	20.7
System of Choice	Preferred localized system in general	68.9
	Preferred non-localized system in general	31.1
Preference in Relation to Question Difficulty	Preferred localized system	48.3
	Preferred non-localized system	48.3
	No preference/no response	3.4
Preference change if glitches occurred in System of Choice	Changed to non-localized from localized	13.8
	Changed to localized from non-localized	3.4
	No change in preference	3.4
	No response	79.4

Table 1. Percentage of student preferences categorized by feedback topics

[2] The basilect variety of English-based Creoles is the most distorted from Standard English.

7. Analysis of Results

The results reported in the previous study using the localized ITS, CRIPSY, confirmed the assumption that cultural interventions do indeed have positive effects on learners and provide empirical evidence in support of localized learning systems. Overall, the students liked the localized system primarily because of the reasons outlined in the earlier studies in [6] namely enriching learning experiences and humour. The use of culture created a familiar setting and it was done in a way that was interesting to the students. A larger percentage of students rated the programming exercises as easier and the instructional feedback/hints as more helpful for the localized system although similar guidance was given in the control system.

In the second study, the majority of students (79.3%) wanted the localization slider. The most common reasons given by students for wanting a slider included: wanting control over the timing and the degree of localization, wanting to change the localization to suit their moods, and wanting to be able to explore the different degrees of localization. An interesting trend in the responses of the students who did not want the slider was their dislike of the localization. Many of them stated that the localization resulted in descriptions that were longer to read and had too much localized language which was confusing. Longer descriptions were indeed the case since the highest density of localization was used in generating content for the study, and therefore the maximum number of lexical insertions and replacements possible for the content was made. This indicates that there is a strong need for the slider since the students essentially wanted to choose their own levels of localisation. Another interesting result is the low student tolerance for software faults in the localized system evidenced by the larger numbers of students who changed their preference to the non-localized system when faults occurred in the localized one. A possible cause could be that students perceived the localized system as being inferior because of the cultural behaviour of the system in using language levels (basilect) typically associated with the less educated in Trinidad and Tobago. This implies that being able to dynamically adjust localization is crucial for system acceptance by students.

8. Conclusion and Future Research

Culture is rapidly becoming an important consideration in the design of educational software firstly because of the increase in the number of users accessing software over the Internet, and secondly because of the sheer diversity in the cultural backgrounds of these users. Conventional learning has often taken place in a localized setting with a teacher guiding one or more students in their search for knowledge. With the advent of the Internet, this traditional setting has changed drastically since students now have access to teachers and educational material from over wide distances. Consequently, these students are exposed to a variety of educational tools, teaching strategies and learning materials which were not developed with their own personal needs in mind. This has dramatic usability implications especially when the mainstream culture for which e-Learning materials are designed clashes with that of the users.

Based on the encouraging evidence established by these studies, the research discussed in this paper demonstrates a practical approach towards developing a localized web-based learning environment. By leveraging research from various fields such as Intelligent Tutoring Systems and culturally-aware instruction, this research shows how some of the complexity of localization can be managed and how aspects of intelligent tutoring can be localized. Empirical evidence indicates that localized systems perform as well as traditional tutoring systems and are potentially superior at creating relaxed, engaging learning atmospheres for the Computer Science programming domain. However care must be taken to ensure that the cultural enhancements match the tolerance level of the student users. Further refinement and improvements are planned for the systems described. A limited amount of cultural automation was undertaken, so expansion of the cultural coverage is necessary. Additional features such as deeper cultural student profiling, adjustable language density and greater tutoring flexibility are also part of the plans intended for this research.

References

1. Blanchard, E.G., Mizoguchi, R., Lajoie, S. P.: Structuring the cultural domain with an upper ontology of culture. In Blanchard, E., Allard, D. (Eds.): The Handbook of Research on Culturally-Aware Information Technology: Perspectives and Models. 179-212. Hershey, PA: IGI Global. (2011)
2. CIA The World Factbook. Available online: [https://www.cia.gov/library/publications/the-world-factbook/geos/td.html] (2012)
3. Gilbert, J.E., Arbuthnot, K., Hood, S., Grant, M. M., West, M.L., McMilllan, Y., Cross, E.V., Williams, P., Eugene, W.: Teaching Algebra Using Culturally Relevant Virtual Instructors. The International Journal of Virtual Reality 7, 1, 21-30 (2008)
4. Henderson, L. Theorizing a multiple cultures instructional design model for e-Learning and teaching. In Edmundson A. (Ed.): Globalized E-Learning Cultural Challenges. 130- 154. London, UK: Information Science Publishing. (2007)
5. McLoughlin, C.: Adapting e-Learning across cultural boundaries: A framework for quality learning, pedagogy, and interaction. In Edmundson A. (Ed.): Globalized E-Learning Cultural Challenges. 223- 238. London, UK: Information Science Publishing. (2007)
6. Mohammed, P., Mohan, P.: The design and implementation of an enculturated web-based intelligent tutoring system for Computer Science education. In I. Aedo, N-S. Chen, D.G. Sampson, J.M. Spector, Kinshuk (Eds.): 11th IEEE International Conference on Advanced Learning Technologies (ICALT). 501-505. Washington, USA: IEEE Computer Society. (2011)
7. Trinidad and Tobago. Available online: [http://en.wikipedia.org/wiki/Trinidad_and_Tobago] (2013)
8. Rehm, M.: Developing enculturated agents: Pitfalls and strategies. In Blanchard, E., Allard, D. (Eds.): The Handbook of Research on Culturally-Aware Information Technology: Perspectives and Models. 362-386. Hershey, PA: IGI Global. (2011)

Impact of a blended ICT adoption model on Chilean vulnerable schools correlates with amount of on online practice

Roberto Araya and Johan Van der Molen

Centro de investigación Avanzada en Educación (CIAE), Universidad de Chile, Chile

roberto.araya.schulz@gmail.com

Abstract. The impact of a blended ICT adoption model in 15 fourth grade classes from 11 vulnerable Chilean schools is analyzed. In this model, twice a week students attend a computer lab where the lab teacher and the class teacher select and assign on line math exercises. The platform contains approximately 2,000 exercises, but it is constantly growing with exercises introduced by teachers. During sessions the system continuously tracks students and detects students that are falling behind, allowing teachers to provide real time support and drive the progression of the entire class as a whole. Students that finish early and with good performances are assigned as part of the support team for the rest of the session. After a year of implementation, the national assessment test SIMCE math on these classes raised 0.38 standard deviations. This is more than three times the historic national improvement in 2011. The statistical analysis shows that the improvement was independent of teacher effects, and correlates with the average number of on line exercises done per student in that year.

Keywords: adoption model, ICT impact, national assessment tests, educational data mining, at risks students

1 Introduction

We study a blended Information Computer Technology (ICT) adoption model focused on ensuring intensive student practice and real time support from teacher and more advanced peers. We implemented this ICT model on several low socioeconomic status Chilean schools with high proportion of at risk students. According to a recent UN report [11], Chilean gross national income (GNI) per capita is 14,987 in 2005 PPP $, Chilean Human Development Index is the highest in Latin America and ranks 40 in the world. Even though gross national income per capita has had an important average annual growth of 3.8 from 1990 to 2012, there is huge school segregation by socioeconomic status in Chile. There is also a big educational inequality measured as interschool standard deviations on a UNESCO math and language tests for third and sixth graders [14], [1], though the educational inequality is generally lower than in other Latin American countries. For example, in the UNESCO math test for third graders, interschool standard deviation was 1.97 the standard deviation obtained if students were randomly assigned, which is one of the lowest ratio of the studied countries. The educational inequality is more pronounced on the urban sector than in the rural sector. According to the last OECD PISA-2009 assessment [16], [17], science and reading performances have been improving and the performance gap in reading between high and low socioeconomic status students has being reducing. From the first UNESCO study [12] to the second UNESCO study [14], the relative position in Latin America of the math performance of Chilean third graders improved. Also the Chilean national math assessment test for fourth graders (SIMCE) shows in 2011 a significant improvement (0.12 standard deviations) and the performance gap has also been reducing (0.22 standard deviations). According to UN [11], fixed broadband internet subscription was 10.5 per 100 people in 2010 in Chile. This is one of the highest subscription rates in Latin America together with Uruguay and Mexico. Fixed and mobile telephone subscribers were 136.2 per 100 people in 2010. In schools, 90.2% of students have computers to use with access to internet [13].

Adoption of ICT generates several benefits in education [18], [19], but it also poses several challenges particularly in vulnerable schools. According to our field experience in vulnerable schools there is a weak ICT infrastructure, poor equipment maintenance, poorly prepared technical support personnel, high frequency of electric supply problems, and instable connection to internet. All of the difficulties were present on the schools where the model was implemented. Besides school infrastructure, there is a much weaker ICT infrastructure at students' homes, less availability of internet access and much weaker access to technical support. Other important difficulties in these schools are a higher proportion of at-risk or behind-grade-level students, higher percentage of students that are struggling with core math concepts, and lower attendance rates than in other schools. Additionally, students' families have less cultural capital, and therefore students are harder to teach. Teacher quality is also generally lower in these schools, since better teachers once detected they receive offers to migrate to higher socioeconomic status schools. All of these conditions pose a much higher challenge in the introduction of ICT and the possibility to have a positive impact.

2 Method

One of the main goals of the implemented model was to provide technological and human resources to increase the amount of student practice on mathematics and also to provide real time feedback for all students, and particularly for struggling students. There is abundant literature reporting the positive effect of practice in ideal laboratory conditions. There is the positive effect on memorization [6], in long term retention [7], in understanding when compared with spending time building concept maps or only studying [3], and the positive effect of practice throughout several sessions [8].

Another important aspect is feedback. There are different types of feedback: immediate versus delayed, simple positive feedback if the answer is correct or negative feedback if it is incorrect, or deeper feedback with suggestions of alternative solutions, feedback aimed at increasing student effort, etc. In a review of dozens of studies [2] concluded that the positive feedback for right answers is better than negative, and that feedback given by video, audio or computer systems is the most effective. There is also evidence of the positive effect of accumulated practice in solving mathematical problems [4]. More practice produces more learning, particularly if practice is mixed with previously learned knowledge. However there is lack of studies of impact outside of laboratory conditions [9], and particularly in low socioeconomic schools. There are several important challenges to implement a strategy based on intensive practice. One of them is motivation. There is the need to spark engagement with math problems throughout the whole year. Moreover to improve the performance of a complete class it is necessary to motivate struggling students to do intensive practice. At the core of the blended adoption model there is an early alert system. At every moment, the system lists students who are having more difficulties. In this way the teachers know in real time which students need personal attention and in what specific exercise. The early alert system also detects if there are exercises that are producing high difficulty to the whole class. This way the teachers can freeze the system and explain the required concepts. The platform is designed to drive the progression of the entire class as a whole, and not to leave students alone. It has facilities to promote the cooperation and support of students that are ahead of their peers. Students that finish early and with good performances can be assigned as part of the support team for the session. The system assigns them to help peers with difficulties. At the same time, students that are being assisted by peers assess the quality of each support event. This information helps the teachers to monitor the quality of the support and the need to teach support strategies to advanced students.

The model was implemented in 11 schools. The first year the system was implemented in the second half of the year, but most of the time was dedicated to solve infrastructure problems. In the second year (2011) the system was running in more normal conditions. All of the 469 students from the 15 fourth grade classes of the 11 municipal schools from the Lo Prado municipality were using the system. Lo Prado is a low socioeconomic district in Santiago. During the year 2011, from end of March to early October the students did 203,782 on line math exercises. After mid-October students did more exercises but they were done after the National test SIMCE, and therefore they will not be considered in the analysis. Around 20% of the exercises were assessment exercises and the rest were exercise with feedback. For every exercise done by a student, the platform records the response time as well as the performance. In the case of an assessment exercise, the system records if the student answered correctly or not, and in the case of exercises with feedback it records the number of attempts to achieve the correct answer.

3 Results

According to What Work Clearinghouse standards of the U.S. Department of Education, evidence of effectiveness [10] of educational interventions is classified into three levels: low, moderate and strong. Evidence is low when it is based only on expert opinion, which is derived from strong findings or from theories in related areas and/or expert opinion. Evidence is rated as moderate when they are made with high internal validity studies (studies whose designs support causal conclusions) but moderate external validity (studies that include a sufficiently broad range of participants to be generalizable), or vice versa, but not both validities simultaneously. At this level, belong the typical quasi-experimental studies (i.e., the control and experimental groups were not randomly assigned) that show the effectiveness of a program, or studies with small samples or groups that are not equivalent at the pretest, or correlational research with strong statistical controls to avoid selection bias or assessments that meet the Standards for Educational and Psychology Testing but the samples are not adequately representative of the population. Evidence is strong if the studies have high internal and external validity. These designs include randomly controlled trials, multisite, large scale, and no contradictory evidence.

We make two types of analysis. First we compare the SIMCE math gains of all the schools of the Lo Prado district with the result of the whole country. We also compare the Lo Prado SIMCE math gains with the results of the urban schools of a very similar neighbor district of the same socioeconomic level. Second, we compare the Lo Prado SIMCE math gains with the Lo Prado SIMCE language gains and SIMCE science gains in the same classes. Given that in each class all subjects are taught by the same teacher, this comparison aloud as to explore the teacher effect and eventually compute if the improvement obtained is independent of the teacher. Therefore, the study reported here could be classified as quasi experimental at best, and then the evidence can be classified as moderate.

One difficulty of analysis is that the SIMCEs scores are not published by student. We only have the average score of each class and the standard deviation of the country. For this reason, in what follows we use the variation of SIMCE for classes. We define the SIMCE variation for a class in a given school as the difference between the SIMCE 2011 score obtained by the class with the SIMCE 2010 average of all classes of the school. It is important to note that the students are different, since they belong to different generations. We are comparing the 2011 score of a class with the average score of the school in the previous year.

The increase of SIMCE math for the municipal schools of the Lo Prado municipality was of 16.27 points, which is greater than the variation of all schools in the country. This increase is obtained from the simple average of the SIMCE math scores of the schools. However, some schools have fewer students, so the simple average of the schools is not the most representative. Weighting proportional to the number of tested students in each course, the variation in SIMCE for math was 19.15 points. This increase is more than three times the historical rise of the country SIMCE math that was 6 points [5], and it is much higher than the increase of 8 points in SIMCE math in the schools with similar socioeconomic classification as 10 of the 11 schools analyzed.

This improvement is an in increase in 0.38 standard deviations. To test if the average improvement of the 15 classes of Lo Prado is higher than the 6 points of the average of the country student's improvements and the 8 points of the average of the country at risk student's improvements, we consider the fact that we have the standard deviation σ of the student performance in the country. From that information we can compute the standard deviation of the average of the score improvement of the 392 students that took the SIMCE test. The standard deviation σ of the student performance does not change from one year to another and σ=50. Assuming independence of the performance between students, the variance of the performance of each class i equals $\frac{\sigma^2}{n_i}$ where n_i is the number of students on the class i, and similarly the variance of the average performance of the school s(i) where the class i belongs to is equals $\frac{\sigma^2}{n_{s(i)}}$, where $n_{s(i)}$ is the number of students on fourth grade at the school s(i). The difference D_i between the score of the class i and the school average on the previous year has a variance equal to $\frac{\sigma^2}{n_i}+\frac{\sigma^2}{n_{s(i)}}=\frac{\sigma^2(n_i+n_{s(i)})}{n_i n_{s(i)}}$. Thus the weighted average score D= $\frac{1}{n}\sum_{i=1}^{k} n_i\, D_i$ of all the k=15 classes of Lo Prado has a variance Var(D)= $\frac{\sigma^2}{n^2}\sum_{i=1}^{k}\frac{n_i(n_i+n_{s(i)})}{n_{s(i)}}$, where n is the total amount of students in the k classes. The null hypothesis is H_0: $\mu = \mu_0$, where the observed mean of D is 19.15 and μ_0=8 is the mean improvement of all the at risk schools in the country (or μ_0= 6 is the mean improvement of all the students of the country). Since Var(D) = 11.1622, the cohen´s d = 3.3719, and then p=0.00074665 for μ_0 = 8. Thus the improvement obtained in Lo Prado is statistically significant. The difference between the country improvement and the average improvement of the 392 students from Lo Prado is statistically significant with a two tailed p-value < 0.0005 and the difference between the improvement of all the country at risk students and the Lo Prado students improvement is statistically significant with a two tailed p-value=0.00074665.

It is very instructive to compare the results of the schools belonging to the municipality of Pudahuel. This is a neighbor municipality and that years ago were both part of a single municipality. In Pudahuel we only consider urban schools. They are similar to those of Lo Prado which all are urban schools. Figure 1 shows the distribution of vulnerability of municipal schools in Lo Prado and the urban municipal schools of Pudahuel. The average of the 11 school´s proportions of at risk students in Lo Prado is 0.80 whereas the average of the 14 school´s proportions of at risk students in Pudahuel is 0.77. The t-test of two sample sizes and unequal variance does not reject that they have different proportions (two tailed p-value= 0.36). In addition, 10 of the 11 schools of Lo Prado are classified as “medium low” socioeconomic status (SES) by the Ministry of Education, and the remaining school is classified as "medium", which is quite similar to what occurs in Pudahuel, where 12 of the 14 schools are classified as "medium low" and the other two as "medium".

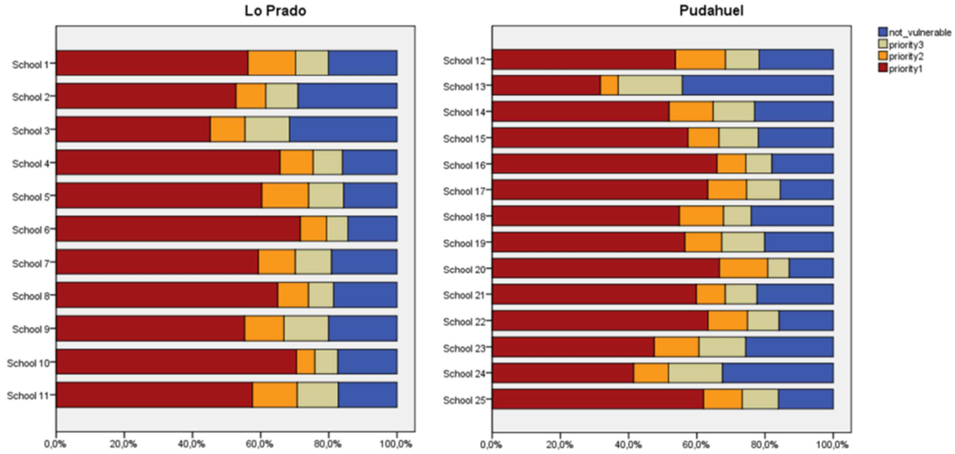


Figure 1: Proportion of at risk students in the schools of Lo Prado and Pudahuel.

In Lo Prado district the increase of SIMCE math schools was 19.15, corresponding to 392 students that took the SIMCE, whereas in Pudahuel the increase was 12.5, corresponding to 790 students. Using again the fact that the standard deviation of the average score improvement of all students on both municipalities can be computed from the standard deviation of student scores, the difference between the average scores however has a p-value = 0.064 unilateral.

There is a possibility that the increase in math SIMCE is due mainly to the contribution of teachers and/or schools. Since in all the classes the teacher teaches all subjects, if the effect is due to specific characteristics of the teacher, then one would expect that classes with increase in SIMCE math then would also have increase in SIMCE language and SIMCE science. Conversely, if there is no clear relation in the increase or decrease across subjects then the increase in mathematics is not only effect of the teacher. In Lo Prado the increase in SIMCE math was 19.15, which is much greater than the increase in SIMCE science SIMCE which was 9.66 (0.19 standard deviations) and the increase in SIMCE language that was -3.35 (0.07 standard deviations). It is important to note that the online exercise strategy was implemented only in mathematics. In language three year ago the municipality implemented a different strategy not using ICT and more recently it implemented another strategy in science but also not using ICT. Furthermore, we tested the statistical independence of changes or if there is any correlation in the 15 classes between the behavior of SIMCE math with SIMCE language and SIMCE science. As shown in Table 1, the Pearson correlation test showed positive and statistically significant correlations between the change in math and language, and between language and science, but not between mathematics and science. These tests ruled out that the possibility that the rise in SIMCE math is due entirely to teacher effect.

Table 1. Correlations between language, math and science variations.

		language_variation	math_variation	science_variation
language_variation	Pearson Correlation	1	,651**	,777**
	Sig. (2-tailed)		,009	,001
	N	15	15	14
math_variation	Pearson Correlation	,651**	1	,334
	Sig. (2-tailed)	,009		,243
	N	15	15	14
science_variation	Pearson Correlation	,777**	,334	1
	Sig. (2-tailed)	,001	,243	
	N	14	14	14

**. Correlation is significant at the 0.01 level (2-tailed).

It is also important to see the effect of the lab teacher in charge of the laboratory. Two lab teachers took charge of the lab on the 11 schools, moving from one to another to attend classes during sessions. The variation in mathematics SIMCE achieved by one lab teacher was 20.14 points which corresponds to 7 classes, while the increase achieved by the other lab teacher was 18 points, corresponding to 8 classes. The t-test for independent samples reveals that we cannot reject the equality of means (N=15 classes). That is, the SIMCE math increase is statistically similar between the two lab teachers.

While the increase in SIMCE math is not explained solely by teacher or lab teacher effect, it remains to be analyzed what may have caused it. As in all schools in the Lo Prado district the model was implemented with one or two sessions per week, it is expected that the amount of exercises attempted had an effect on learning. Figure 2 shows the variation of SIMCE math in the 15 classes of the 11 schools of the district, as function of the number of online exercises with immediate feedback between end of March and early October 2011. In total, there were 185, 413 such exercises. Exercises with errors in their statements were discounted. Also were not counted exercises answered by a student in less than three seconds and in less time than the average response time minus two standard deviations. As seen in the chart there is a positive relation between number of exercises and increase in SIMCE math. The increase is of 15.3 points in SIMCE math per 100 extra exercises per student performed online. This means that for every 12 additional exercises done per month per student, the SIMCE math increases 15.3 points (0.306 standard deviations). The result is statistically significant with a p-value of 0.029.

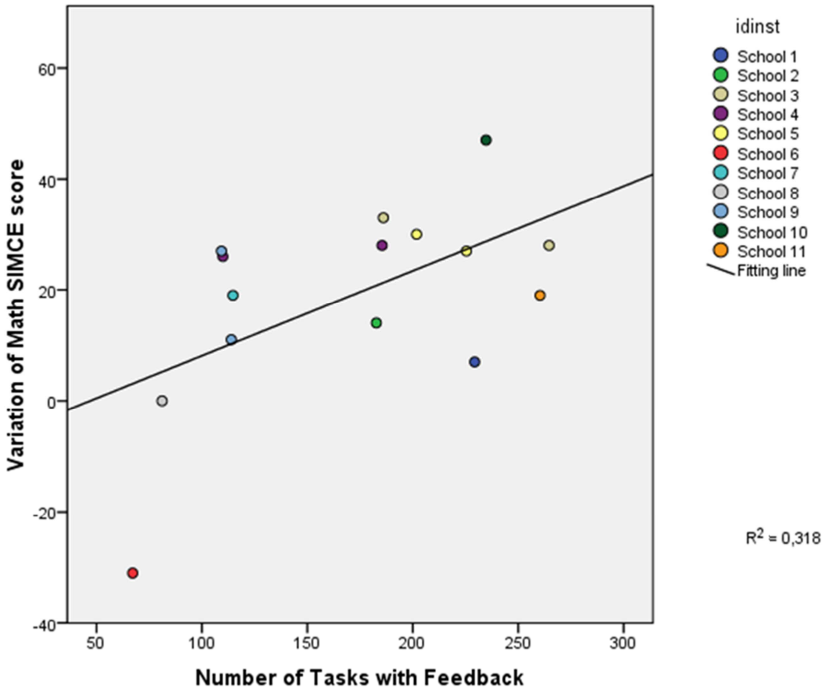


Figure 2: Variation on the 2011 SIMCE Math score of each class (N=15 classes) in function of the number of on line exercises with feedback made per student during the year from end of March to early October 2011.

4 Discussion

We have measured the impact of the adoption of a blended model focused on ensuring intensive student practice and on providing real time support from teachers and peers in all the 11 public schools in a low socioeconomic district. There are 203,782 records corresponding to the trace of on line math exercises done in 2011 by the 469 students in all the 15 fourth grades classes on these schools. The results are very promising and encouraging. First, there is a real measurable impact. The effect has been obtained not only by internal tests, but using the results of the national assessment test SIMCE math. This is a nationwide test, designed and administered by a completely independent team belonging to the Ministry of Education. The impact is very big. The increase was 0.38 standard deviations. This is more than three times the big and historic country wide increase on the SIMCE math of the year 2011. These results are in addition to differences in teachers. Moreover, there is a clear correlation between improvement and intensity of use of the ICT model. Classes that did more exercises with immediate feedback per student in the year they increased more their SIMCE math scores. This result agrees with studies in laboratory conditions, but now the results are obtained after a year of implementation in real schools and with low socioeconomic status students.

Acknowledgements: Conicyt CIE-05 Centro de Investigación Avanzada en Educación, and Fondef TIC EDU TE10I001.

References

1. Araya, R. & Gormaz, R. (2012) How come educational inequality in Cuba is much higher than in Chile? CIAE publication http://www.ciae.uchile.cl/index.php?page=view_publicacion&id_publicaciones=252
2. Hattie, J. & Timperley, H. (2007) *The Power of Feedback, Review of Educational Research. Vol 77, No. 1, pp. 81-112.*
3. Karpicke, J. & Blunt, J. (2011) Retrieval Practice Produces More Learning than Elaborative Studying with Concept Mapping. *Science, 20 January.*
4. Mayfield, K. & Chase, P. (2002) The effects of cumulative practice on mathematics problem solving. *Journal of Applied Behavior Analysis, 35, 105-123*
5. Ministerio de Educación (2012) *Resultados para Docentes y Directivos.* http://www.simce.cl/fileadmin/Repositorio_4B/2011/Docentes_y_Directivos/DOC_DIR_2011_4B_RBD-10089.pdf
6. Pyc, M. & Rawson, K. (2010) Why Testing Improves Memory: Mediator Effectiveness Hypothesis. *Science, 15 October 2010: 335 DOI: 10.1126/science.1191465*
7. Roediger, H. & Karpicke, J. (2006) Test-Enhanced Learning: Taking Memory Tests Improves Long Term Retention. *Psychological Science.*
8. Rohrer, D. & Pashler, H. (2007) Increasing Retention without Increasing Study Time. *Current Directions in Psychological Science*
9. U.S. Department of Education. (2009) *Effectiveness of Reading and Mathematics Software Products Findings from Two Student Cohorts.* Institute of Education Sciences National Center for Education Evaluation and Regional Assistance. February 2009
10. U.S. Department of Education. (2009) *Using Student Achievement Data to Support Instructional Decision Making.*
11. UNDP (2013). *Human Development Report 2013. The Rise of the South: Human Progress in a Diverse World.*
12. UNESCO (2001) *Primer estudio Internacional Comparativo sobre Lenguaje, Matemática y Factores asociados para alumnos de tercer y cuarto grado de Educación Básica.*
13. UNESCO (2008) *A view inside primary schools: A world education indicators (wei) cross-national study.* Technical Report.
14. UNESCO (2010) *Factores asociados al logro cognitivo de los estudiantes de América Latina y el Caribe.* Santiago.
15. UNESCO (2013) *A View Inside Primary Schools. A World Education Indicators (WEI) cross-national study.*
16. OECD (2010*) PISA 2009 Results: Overcoming Social Background: Equity in Learning Opportunities and Outcomes. Volume II*
17. Ministerio de Educación (2011) *Resultados PISA 2009 Chile*
18. Wims, P. and Lawler, M. (2007) Investing in ICTs in educational institutions in developing countries: An evaluation of their impact in Kenya. *International Journal of Education and Development using ICT* [Online], 3(1), http://ijedict.dec.uwi.edu/viewarticle.php?id=241

19. GeSCI (2011) *ICT, Education, Development, and the Knowledge Society*. Retrieved from http://www.gesci.org/assets/files/ICT,%20Education,%20Development,%20and%20the%20Knowledge%20Society(1).pdf

Children Creating Pedagogical Avatars: Cross-cultural Differences in Drawings and Language

Melissa-Sue John[1], Ivon Arroyo[1], Imran Zualkernan[2], Beverly P. Woolf [3]

[1] Social Sciences and Policy Studies, Worcester Polytechnic Institute
[2] Computer Science and Engineering, United Arab Emirates
[3] School of Computer Science, University of Massachusetts Amherst

iarroyo@wpi.edu, mjohn@wpi.edu, izualkernan@aus.edu, bev@cs.umass.edu

Abstract. This research identifies cultural differences among children's drawings especially as related to their drawings of avatars for instructional software. We invited children to draw characters and textual messages within an instructional game, as a way to establish their expectations of pedagogical avatars. We were interested in both the appearance and language of the characters of different nationalities. We describe an experiment that evaluated cultural differences in children's drawings. We analyzed drawings produced by 57 children aged 7-10 from four countries and discovered several main effects. Specifically, a significant main effect was found for a child's nationality and gender in predicting the emotion, formality of language, and use of "polite" or nice language. Girls generally expected more details in the hair, skin and facial hair of their characters and drew more emotions (positive) into their characters. Additionally, Pakistani and Argentine boys drew more details and more head coverings than did other children. Girls from Pakistan drew fantasy figures, rather than realistic figures and did not draw headscarves on their characters. The level of detail expected in the characters varied by country.

Keywords: Developing World, Pedagogical Agents, Children's Drawings, Localization

1 Introduction

Pedagogical agents used within adaptive learning environments have provided great benefit for learners as indicated by research over the last few decades (Lester, et al., 2004, Blair, Schwartz, Biswas, & Leelawong, 2006; Biswas, Schwartz, Leelawong, Vye, & TAG-V, 2005). Pedagogical agents are effective tools to support student learning; they provide motivation for learning and promote positive affective states (Arroyo, Woolf, Royer, & Tai, 2009). Results have shown that students are extremely sensitive to the appearance and gender of the characters reacting in different ways, and being more or less productive depending on the character's appearance. In a series of studies, students responded more positively when the gender of the character matched the gender of the student (Arroyo et al., in press).

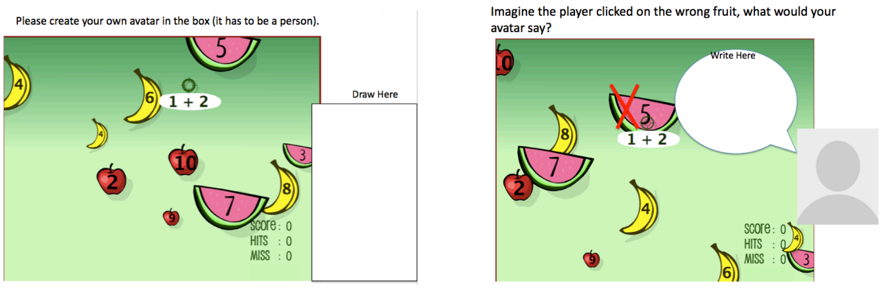
When considering the migration of educational systems and learning environments to other countries, it is unclear whether pedagogical agents would work in a similar way for students of developing countries. Should agents mimic the gestures and even dress codes of students in different countries, or is this localization effort beyond translation unnecessary? Are there differences in the style of language that pedagogical agents should use to communicate with students of different nations?

As a way to measure ecological validity, we decided to carry out an experiment that "taps into" children's minds and their expectations for pedagogical agents. We asked students to create their own pedagogical agents or avatars that would guide them through a mathematics learning game. The following article describes an experiment across four different countries in different continents, summarizes results and draws conclusions about the way to move forward to identify children's cross-cultural differences in expectations for a helpful avatar.

1.1 Background and Related Work

Having children draw as a way to mirror what is in their minds is a common technique used in psychology. Research into children's drawings has focused on three main areas: (a) the internal structure and visual realism of children's depictions (e.g., Cox, 1992); (b) the perceptual, cognitive, and motor processes involved in producing a drawing (e.g., Freeman, 1980); and (c) the reliability and validity of the interpretation of children's drawings (e.g., Hammer, 1997). Very young children produce simple scribbles, and later demonstrate representational intentions. With maturation and increased dexterity, children draw objects as they are known rather than as they are actually perceived.

Drawings of the human figure can also reflect a child's social world. La Voy and colleagues (2001) explored the idea that children from different cultural backgrounds may represent cultural differences in their drawings, because culture permeates a child's representations of people. Differences across nations indicated that American children drew more smiles than Japanese children, whom in turn drew more details as well as larger figures (La Voy et al., 2001). Similarly, Case and Okamoto (1996)



Please create your own avatar in the box (it has to be a person).

Imagine the player clicked on the wrong fruit, what would your avatar say?

Figures1-2. A simple addition math game for younger children. Children were invited to supply a drawing for an avatar (left) and then to provide the responses the avatar might provide when the student player chose the wrong mathematics answer.

showed that there are cultural differences between Chinese and Canadian children's drawings. These findings suggest that children's drawings not only reflect representational development but a child's understanding of self and culture as well.

Having students draw characters and games, as a way to tap into their minds and establish their expectations of pedagogical characters and games is an increasingly common technique, and has particularly been implemented for learning systems/games for mathematics education. For instance, Grawemeyer and colleagues (2012) managed to have participants within the autism spectrum express and externalize their individual ideas for an educational pedagogical agent for a mathematics educational game, and to combine their individual ideas with the ideas of others in a small group. Students created their own designs and also studied other students' drawings, eventually creating a common prototype.

The outcome of one of the small groups was quite different from the norm: these children with autism designed characters, such that the student would be sitting at the back seat of a car, being able to view two avatars sitting in the front seat, from the view of the person in the back seat. Instead of showing the avatar facing forward and expressing emotions through its facial expressions, as has commonly been done in the past, the avatars (shown from the back) would have a conversation about the student's learning and progress, as children might interact with their parents when traveling at the back seat of the car. Thus these students with autism expressed their own distaste for talking directly to at people or looking into their eyes. It is assumed that an avatar designed for a typical student would promote better communication if it looked directly at the student.

Other studies have invited children to design and draw full math games, which generally included characters, human or not. For instance, Kafai (1996) invited fourth grade children to design mathematics games for younger children. Her study, identified important gender differences in the design of games. In general, boys were more likely to use fantasy locations in their games (instead of real life locations, such as a sky slope), and also were more likely to have the presence of evil characters, or the idea that an avatar would fight some evil force.

2 The study

Our study involved children invited to draw characters, avatars or pedagogical learning companions to keep student players company as they used a game to learn mathematics. The goal was not to ask for complex representations, but instead, and similar to La Voy and colleagues (2001) to explore cultural differences that are important to understand for authors of creating pedagogical avatars.

,

Children from North America Argentina, Pakistan, and Jamaica, aged 7-10 were asked to draw characters they thought would help younger support as they played a mathematics game for younger children. Children were given a printed package of 6 pages. On page 1, students were told "Help us design this math game! We are designing computer based math games for younger children. Can you help us?" On the second page, a screenshot of a simple addition math game, shown in Figure 1, where student players would click on the fruit with the right answer is shown and at the top reads "This is a picture of a math game. In this game, children will learn to add. Using the mouse, they have to click on the fruit with the right answer." The children were invited to provide a voice for their avatar by providing a response that the avatar might produce in response to a student player's incorrect answer, see Figure 2. And finally parents and teachers were instructed to complete the student demographics (age, ethnicity, nationality and gender).

We obtained drawings from 57 children from North America (14), Pakistan (11), Jamaica (18) and Argentina (14). Of these children, 30 were girls and 22 were male, mean age was 8.19 (SD = 1.42). We were interested in both the appearance and language of characters developed by these students of different nationalities.

Properties of avatar

1. Realism (Human / Fictional)
2. Gender (F / M / Unspecified)
3. Age (Child / Teen / Adult / Unspecified)
4. Details (+ 1 for each of these: body, eyes, nose, mouth, dimples/freckles, ears, teeth, hair, facial hair, head-covering, clothing, shoes, accessories, toys, skin-coloring)
5. Affect (Happy / Neutral / Sad / Angry)

Voice of avatar

7. Tone of incorrect answer (Polite/encouraging or Direct/Straightforward or rude/aggressive/discouraging)
8. Formality of incorrect answer (formal/neutral/informal)
9. Tone of correct answer (Polite/encouraging or Direct/Straightforward or rude/aggressive/discouraging)
10. Formality of correct answer (formal/neutral/informal)

Characteristics of Participant

11. Language spoken/written (English/Spanish/Pashto)
12. Videogame exposure (Do you play videogames?)
13. Have you ever **used** an avatar?
14. Have you ever **created** an avatar?
15. Age Student
16. Gender: Female (1) Male (2)
17. Ethnicity
18. Nationality

Table 1. Features of the study to analyze cultural characteristics of children's drawings. Properties of the avatar, voice of the avatar, and characteristics of the student, were analyzed to explore cultural differences.

3 Results

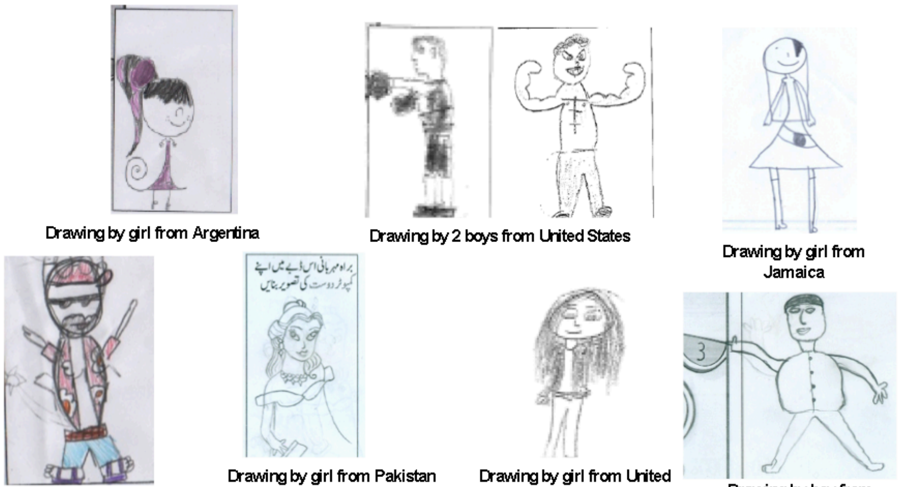
Although children were asked to create math avatars that looked like people, children came up with humanoid and non-humanoid images. In one study in particular, it was not clear that students had understood that we meant "characters that look like humans". Thus, for the purpose of our analyses, we only coded humanoid images (see Figure 3).

Two different human coders analyzed the pictures and messages to respond to correct/incorrect answers from student players. They coded the variables described in Table 1. Because many of these metrics might be somewhat subjective, we had two coders separately. After coding was done, we computed Kappa to analyze agreement between the coders. Whenever a variable had a Kappa value less than 0.5, we reconsidered the variable and came up with a new coding scheme. The variable was recoded and the process repeated. Variables with very low Kappas were dropped from the analysis (e.g., age of the avatar). We then carried out Analysis of Variance with the variable of interest, and nationality, gender-child as fixed factors, with age of child as a covariate. In the case of discrete variables, we ran cross-tabulations and Chi-Dquare tests. Results indicate the following.

Gender of Avatars. A significant difference was found for child's gender ($\aleph^2$ =38.9, p<0.001) and gender of the avatar, showing that most children drew characters of their same gender. No significant differences were found for nationality. Only a minority of children drew characters of unidentifiable gender.

Realism of Avatar. A significant interaction effect between gender of the student and nationality (F=3.9, p<0.015) showed that Pakistani girls drew more fantasy characters than did children from other countries, or than boys of the same country (see Figure 4).

Level of Detail. A significant main effect was found for a child's nationality in predicting level of detail of the characters (F=3.6, p<0.02). Students from Pakistan and Argentina drew more details than did children from the United States or Jamaica, regardless of their gender, two more features from Table 1 (see *4. Details*) on average. Further analyses showed differences in the amount of head-coverings, particularly drawn by boys in general (F=13.6, p<0.001), and for Pakistani and Argentine boys in particular (F=13.6, p<0.12), who drew more headcoverings. While we expected girls



Drawing by girl from Argentina

Drawing by 2 boys from United States

Drawing by girl from Jamaica

Drawing by boy from Argentina

Drawing by girl from Pakistan

Drawing by girl from United States

Drawing by boy from Pakistan

Figure 3. A selection of avatars drawn by children in different cultures as companion for a proposed math game.

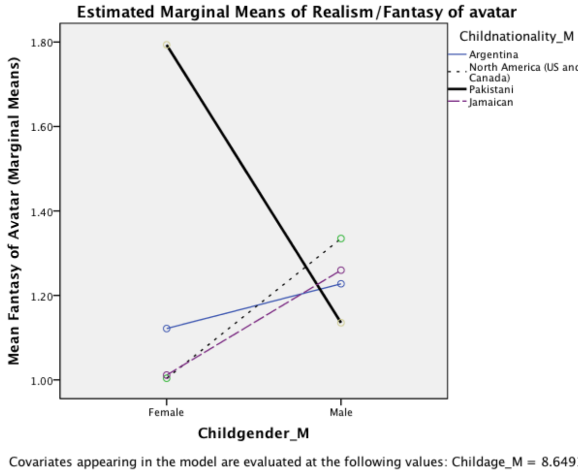


Figure 4. Girls from Pakistan drew more fantasy characters than did other children

from Pakistan to draw headscarves, they did not –in fact they tended to not draw images of real people but drew fantasy figures from other cultures such as princesses. The head accessories that boys drew were actually hats.

Another difference had to do with the drawing of clothes –children from the United States drew the least detailed clothes on their avatars (F=3.5, p<0.01). At the same time, students from Jamaica drew more hair on their characters' heads, and students from Argentina drew the most facial hair on their avatars. Meanwhile, girls in general drew sigificantly more hair on their avatars, both on the characters head (F=11.17, p<0.02) and more facial hair details (note this included eye-brows, eye-lashes, moustache, etc.) (F=8.2, p<0.001). Girls also drew more details on the skin (e.g. freckles, dimples, tatoos, etc.) (F=4.5, p<0.04). No significant differences were found in the amount of accessories used, the kind of accessories, nor in the presence of shoes, noses, eyes, nor bodies. Most students drew all of these, mostly full-bodied avatars instead of heads, and the amount of accessories did not have a consistent differential pattern across nations or genders.

Emotions. A significant main effect was found for emotions expressed by avatars for girls and boys. Girls across nations were more likely to draw avatars with happy faces, with boys evenly split

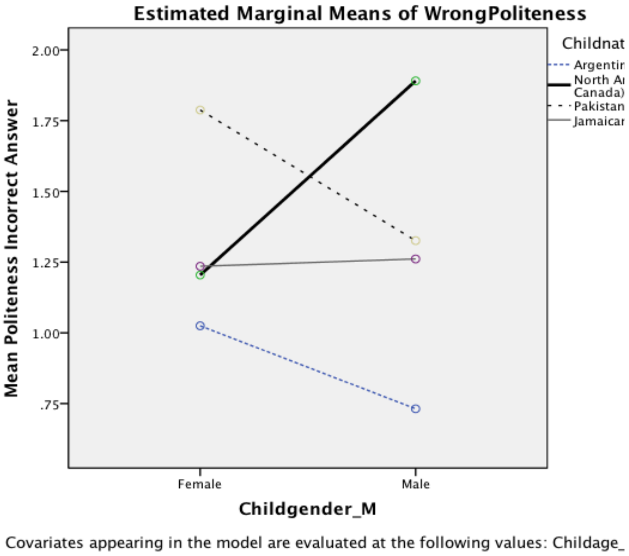


Figure 5. Gender differences in politeness of the avatar's response to incorrect answers from student player

between happy and neutral faces (gender effect, F=9.8, p<0.003) and a minority of children drew angry/agressive emotion in their characters, 5% of all students, all three were boys instead of girls.

Formality of Language. A significant main effect was found for a child's nationality predicting the formality of language for the avatars response to incorrect or correct answers from student players in the game (F=9.7, p<0.001). Students from the United States used more informal language than did students from Jamaica and Argentina (e.g. nope for "no", awesome), and children from Argentina used the most formal language (i.e. least informal language) in their answers than Jamaica, United States and Pakistan.

Tone of Language. Significant effects were found across countries for the avatars' answer after an *incorrect* answer from a student player, where as no significant differences existed for having the character express a response after a player's *correct* answer. A significant main effect for nationality (F=3.3, p<0.03) showed that students from Argentina used the least "polite" language as compared with students from other countries (e.g., least use of words such as *sorry*, *please*, *thank you* etc.), with children from the United States and Pakistan using the most polite language. Interestingly, there was an interaction effect between gender of the child and nationality (F=2.9, p<0.05), which indicated gender differences in the tone of the avatar's response to incorrect answer for children of different genders. Actually, boys' avatars from the United States used more polite language than girls' avatars from the same country, despite the fact that the appearances of U.S. boys' tended to be more aggressive than girls' (see examples in Figure 3 drawings); the reverse happened for Pakistan, where girls' avatars used more polite language than boys' (see Figure 5).

4 Discussion

Some research articles have claimed that children's drawings are a mirror to children's minds (Cherney et al, 2006). In light of this, what do these results imply in terms of the creation of pedagogical agents, and the translation of adaptive learning systems to fit new countries, after some important differences in the look and conversation of children's pedagogical agents? If we consider that what children draw is what they expect, value, and desire, the findings suggest that children, regardless of country, expect characters to be of their same gender. This is consistent with our prior findings (Arroyo et al, 2013), which indicated that matching the gender of the student with the character's gender led to improved affective, behavioral and learning outcomes, such as engagement and reduced frustration. Girls also expect more details in their character's hair, skin and facial hair. Boys might want to have more head coverings, particularly hats. Also, girls from Pakistan might prefer fantasy figures instead of figures that depict themselves. Lastly, the fact that girls in general drew more emotions (positive) on their characters could suggest an expectation of girl's avatars to emote and act affectively –however, this needs to be examined further.

It does seem important that the level of detail expected in the characters will vary by country. Children from Argentina and Pakistan might expect more level of detail in their characters than do students of the United States or Jamaica, e.g. clothes and hair. Meanwhile, differences across countries are especially marked in the *kind* of language to be used when the characters talk, specifically when student playes produce incorrect answers, with Argentine children apparently expecting the least politeness. Expectations of politeness and niceness of the language can be explained by cultural differences. People in Argentina are very straightforward in their dialog (similar to European countries such as France, Italy or Spain) and do not excuse themselves so much in their daily interactions. This is something that needs to be examined when designing characters that communicate with students, even if the communication is in the form of text and not voice. This would potentially argue against a mere translation from English to Spanish, where such polite words might show up. Differences in formality of the language between Argentina and the United States could be explained by the fact that the language might not lend itself to informal distortion of words such as "nope".

5 Conclusions and future work

Large differences were observed in children's design of pedagogical agents across a variety of dimensions, but probably in different areas than we had originally expected. . Differences were present across countries, across gender, and across country and gender. Main differences were in language of incorrect answer across countries, and in the look of characters, both across countries and genders. These differences span across the visual appearance of pedagogical agents as well as in the language used to communicate to student players. From a methodological point of view, having children design pedagogical agents by having the freedom to draw and create, can act as a mirror to their minds and help researchers to externalize their expectations.

The main limitation of this study has to do with the total amount of subjects available, which is not representative of different socio-economic levels of each country, as well as a lack of representation in terms of ethnicities in each country, Future work will consist on a larger study, with a much larger number of students.

Acknowledgements

This research was funded by an award from the National Science Foundation, NSF # 1109642, REESE, Personalized learning: strategies to respond to distress and promote success, Ivon Arroyo (PI), with Beverly Woolf and Winslow Burleson. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Arroyo, I., Burleson, W., Tai, M., Muldner, K., Woolf, B.P. (in press) Gender Differences In the Use and Benefit of Advanced Learning Technologies for Mathematics. Journal of Educational Psychology

Arroyo, I.; Woolf, B.P., Cooper, D.G., Burleson, W., Muldner, K. (2011). The Impact of Animated Pedagogical Agents on Girls' and Boys' Emotions, Attitudes, Behaviors and Learning. ICALT 2011, IEEE's International Conference on Advanced Learning Technologies, Athens, GA.

Blair, K., Schwartz, D.L., Buswas, G., & Leelawong, K. (2006). Pedagogical agents for learning by teaching: Teachable agents. Educational Technology & Society, Special Issue on Pedagogical Agents.

Biswas, G., Schwartz, D. L., Leelawong, K., Vye, N., & TAG-V. (2005). Learning by teaching: A new paradigm for educational software. Applied Artificial Intelligence, 19(3).

Case R., & Okamoto, Y. (1996). The role of central conceptual structures in the development of children's thought. Monographs of the Society for Research in Child Development, 246(61), 1–2.

Cherney, I. D., Seiwert, C. S., Dickey, T. M. & Flichtbeil, J.D. (2006) Children's drawings: A mirror to their minds. Educational psychology: An International Journal of Experimental Educational Psychology, 26(1).

Cox, M. V. (1992). Children's drawings. Harmondsworth, UK: Penguin

Grawemeyer, B., Johnson, H., Brosnan, M., Ashwin, E., Benton, L. (2012) Developing an Embodied Pedagogical Agent with and for Young People with Autism Spectrum Disorder. Intelligent Tutoring Systems 2012. Springer.

Freeman, N. H. (1980). Strategies of representation in young children: Analysis of spatial skills and drawing processes. London: Academic Press.

Hammer, E. F. (1997). Advances in projective drawing interpretation. Springfield, IL: Thomas

Kafai, Y. B. (1996). Gender differences in children's constructions of video games. In Patricia M. Greenfield & Rodney R. Cocking (Eds.), Interacting with video (pp. 39–66). Norwood, NJ: Ablex Publishing Corporation.

La Voy, S. K., Pederson, W. C., Reitz, J. M., Brauch, A. A., Luxenberg, T. M., & Nofisnger, C. C. (2001). Children's drawings: A cross-cultural analysis from Japan and the United States. School Psychology International, 22, 53–63.

Lester, J., Branting, K., and Mott, B., (2004) Conversational agents. The Practical Handbook of Internet Computing

Comparing Paradigms for AIED in ICT4D: Classroom, Institutional, and Informal

Benjamin D. Nye

Institute for Intelligent Systems, University of Memphis
Memphis, TN 38111
benjamin.nye@gmail.com

Abstract. The landscape of technology in the developing world is changing significantly, primarily due to the rapid expansion of mobile computing devices. These changes make it important to re-evaluate practices for internet and communications technology for development (ICT4D). This paper examines three alternative paradigms for educational technology in the developing world: traditional classroom systems, institution-wide systems, and informal learning systems. The advantages and disadvantages of each paradigm are considered in terms of barriers to adoption at the student, teacher, and institutional level. Consideration is also given to educational technologies that serve as models for each type.

Keywords: Educational Technology, Barriers to Adoption, Informal Learning, Ubiquitous Learning

1 Introduction

As access to Information and Communications Technology (ICT) expands through the developing world, educational technology has the potential to play a pivotal role for supporting development. However, successful paradigms for incorporating ICT into developing world education are less clear. Educational technology in the developing world has an uneven history that includes numerous wasted investments in underutilized computers and limited learning benefits (Patra et al., 2007; Woolf et al., 2011). Moreover, the landscape of ICT in the developing world is changing drastically due to the rise of mobile handsets and wireless Internet access (International Telecommunication Union, 2012). These changes offer new opportunities, but also present new obstacles.

Research on advanced intelligence in education (AIED), such as adaptive learning systems, intelligent tutoring systems, and computer-supported collaborative systems, needs to outline the tradeoffs between different application contexts (e.g., classroom, institution-wide, and informal) to help select appropriate system designs. In this paper, these tradeoffs are framed as factors that mediate adoption of ICT, as indicated in Table 1. These factors are based on known barriers to information and communications technology that were identified from recent review papers (Gulati, 2008; Lowther et al., 2008; Bingimlas, 2009). Different paradigms have advantages and disadvantages for each factor.

Table 1. Factors Impacting Adoption of Educational Technology

System	Requirement/Possible Barrier	Description
Learner	Basic ICT skills	Computer literacy and familiarity with basic interfaces
	Independent access to ICT	Web access or computing outside of school
	Motivation to use ICT	Student interest and persistence in use
	Peer support	Peer help or collaboration
Teacher	Basic ICT skills	Computer literacy and managing applications
	Beliefs about utility of ICT	Values and expectations for an ICT design
	ICT-integrated curricula	Pre-made curricula and syllabi that incorporate an ICT intervention
	Match to pedagogical views	Match of teacher pedagogy to an ICT design
	Peer support	Communities of practice and peer views
	Time constraints	Class and preparation time available
	Training (e.g., in-service)	Training with a given ICT design
School or Institution	Administrative support	Administrative needs, reactions, and leadership toward ICT use
	Curriculum flexibility	Flexibility to modify teaching to use ICT
	ICT hardware availability	School web access and computing hardware
	Technical support	Technical staff to set up and maintain ICT
	Internet reliability	Stable, reliable internet connections

This paper considers three paradigms that have shown promise in the developing world, discussing successes and potential challenges. These paradigms will be framed in terms of the context where they are utilized: under *classroom* control, around the entire *institution* (e.g., through a central learning management system), or outside the educational system in an *informal* learning context. While these are not the only approaches (nor are they exclusive), each offers distinct strengths and challenges. Each paradigm will be briefly discussed, with attention to the barriers to sustainability noted in Table 1 and also to promising implementations that embody each approach.

2 Traditional Paradigm: ICT Under Classroom Control

The traditional paradigm for educational technology in the developing world has been classroom-centric (Gulati, 2008). The typical design sets up classroom computers or shared computer labs with educational software. In this context, educational technology is a tool that teachers use to improve learning for students. Classroom-based tools are typically tailored to domain (e.g., Algebra I) and require less flexibility than a general learning management system (LMS).

Classroom-centric ICT has many advantages when compared to other approaches. First, the classroom setting gives the teacher a significant degree of control over students to mandate and manage the use of the system by students. In a classroom setting, basic ICT skills are not typically a blocking issue as students often learn controls quickly and students with more advanced ICT skills may even help the teacher (Gulati, 2008; Ogan et al., 2012). As such, the high availability peer support mitigates deficits in basic ICT skills. Students also do not need to own personal computing devices. The motivation of students, while still important, is less critical than in other contexts. Research has found that liking a system does not necessarily correlate with learning gains, provided students still use the system as intended (Moreno et al., 2002). In a classroom, most students will do assigned work even if they do not find it interesting.

Second, the primary buy-in occurs at the teacher level. At least for initial evaluations, this mitigates many barriers related to teachers. Given that teachers have very different attitudes to technology (Lowther et al., 2008), the ability to pair up a system with technologically-receptive teachers greatly increases the likelihood of successful usage. Teacher beliefs about ICT, match to pedagogical views, and these teachers' basic ICT skills are likely to be better than average. One barrier not mitigated by this approach is peer support, as few teachers will be using the system. Additionally, scaling up to widespread use will hit these barriers once the supply of early-adopters is exhausted. Persuading uninterested teachers to adopt technology is unlikely, unless institutional entities encourage its use. So then, while this paradigm is useful for pilot testing and establishing a foothold, there may be limits to its scale.

The clear point of failure for a classroom-centric approach is institutional factors. If buy-in is primarily at the teacher level rather than the administration level, there is no assurance that the larger institutional context will offer a sustainable environment for that educational technology. If educational technology is a low priority, teachers may be pressured to focus on other matters and technical support may be unavailable. Inflexible mandatory curricula may also make it impossible to work technology into classrooms. Alternatively, curricula dedicated to computers may focus exclusively on digital literacy (e.g., learning about computers) rather than using computers to learn a broader range of topics.

Most importantly, ICT hardware depends on financial support. Investment in computers must be made at the institutional level, but developing world schools often lack the funding to support heavy investment into purchasing, managing, and replacing hardware. Low ratios of students to computers can make meaningful computing curriculum infeasible. Accessing and financing reliable Internet may also be out of the control of the school system. Many developing world areas still have unreliable electrical and Internet infrastructure, which can easily fail and derail any instructional plan relying on web connectivity (Woolf et al., 2011). So then, the primary barriers to traditional classroom ICT are at the school and institutional level. Thankfully, strong focus has been placed on overcoming hardware barriers for ICT in schools. Irregular electricity can be mitigated by using laptops, as their batteries make them immune to short power losses. Irregular Internet can be sidestepped by installing from disk media or only depending on Internet infrequently, rather than during classroom time. Pilots of Cognitive Tutor in Latin America installed software on desktops and did not note significant roadblocks due to the unreliable Internet available (Ogan et al., 2012). By implication, web-based tutoring portals are poorly-suited for the developing world classrooms. This is unfortunate, since educational technology in the developed world has moved strongly in this direction.

Two approaches have been used to overcome hardware barriers: cheaper devices and shared computing. The One Laptop Per Child program spearheaded the “cheaper hardware” approach, driving down the base cost of computers overall (Patra et al., 2007). However, this approach encountered two problems. First, even with lower costs, many schools cannot afford a laptop for every child. Second, studies on ICT interventions in the developing world find that students *prefer* to share computers (Ogan et al., 2012). As such, a number of systems have adapted to this landscape and offer one mouse or keyboard per child (Alcoholado et al., 2012; Brunskill et al., 2010), collaborative turn-taking, and other methods of individual input into a shared learning environment such as mobile devices (Kumar et al., 2012) or wireless clickers (Zualkernan, 2011). Individual inputs are inexpensive compared to computers, greatly reducing hardware costs. Additionally, these techniques complement cheaper computers since they have a multiplier effect. Computer sharing also offers greater pedagogical flexibility, since interactions with other students enable social constructivist designs that would be difficult in a single-user system.

MultiLearn+ offered one model for such a multi-input system, presenting a math game split into four quadrants on a laptop screen and supplying each student a numeric keypad (Brunskill et al., 2010). To prevent dominance by a single student, MultiLearn adapted the difficulty of questions based on student performance. This system relied on installed software, with no Internet component. At present, a laptop with educational technology designed to be shared by four or five students may be the best model for ICT in a primary or secondary school classroom. Such a system might use Internet to update the system, but cannot assume Internet will be available during a classroom session. While significant work has been done in this area, there are still many questions over the relative advantages of different presentation devices (e.g., laptop screens, projectors, voice narrative) and input devices (e.g., mice, keyboards, voice recognition, game controllers, clickers/remotes). In particular, shared mobile computing might be a transformative technology in the future. For example, Kumar et al. (2012) presented a mobile learning tutoring system based on voice recognition and suggested the potential for shared computing through voice identification. While this particular paradigm may encounter technical hurdles, computer sharing for mobiles is an important avenue that needs further research.

3 Institutional Paradigm: ICT Around the School

In a related paradigm, the institution controls a learning management system from the top down. The institution may be a school, district, or even a national system. Learning management systems (LMS) primarily provide a container and delivery platform for static media, though assessments, adaptive learning systems, collaborative systems, or tutoring systems may be incorporated. These systems can support both traditional and online classes. Worldwide, this is more common within higher education. Ubiquitous systems, which connect a variety of devices to a central system, also require an institutional paradigm.

Institutionally-centered systems have similar pros and cons with respect to student barriers, since an instructor usually guides a group of students. One advantage is that, since students interact with a shared central system, remote peer support is possible (e.g., a forum, Wiki, or social media). Unlike classroom-centered systems, institutional systems typically require each student to own a personal computing device. This is because primary use cases of LMS and ubiquitous learning are web-based homework and remote collaboration. However, teachers are the most affected by this paradigm, who will often need to redesign their curricula to fit the system. While an opt-in single classroom paradigm hides teacher barriers by excluding the most resistant or inadequately-prepared teachers, institution-wide adoption hits these barriers head-on. Institution-level barriers are also still an issue. While buy-in by the institution should increase administrative and technical support, hardware costs remain an issue. Since an LMS requires both servers and personal computing hardware, centralized institutional paradigms are more hardware intensive and more costly as a result.

A few designs have attempted to overcome these limitations. EDUCA, a ubiquitous learning platform, provides an LMS and tutoring system capabilities that can be accessed asynchronously over the web through a desktop or a mobile device (Cabada et al., 2011). Entire learning modules are downloaded to the mobile device, as well as an adaptive system for personalizing learning. Since mobile Internet is more prevalent than wired Internet in the developing world, this helps students access the system without a home computer. However, as Mexico is an “emerging market,” this approach still may not translate to less developed countries with worse wireless infrastructure. An alternative approach enabled mobile devices to communicate with the school network over mobile web or through “learning pills” transferred to the student’s phone during class over Bluetooth (Munoz-Organero et al., 2012). However, both approaches require the student to own a web-capable phone and passes these costs down onto students. This approach seems better suited to higher education, where students can be responsible for the costs and ownership of mobile computing devices. However, as mobile web capability becomes commonplace, ubiquitous paradigms may also become relevant for primary and secondary education. In either case, any learning management system or large-scale institutional system for the developing world must support mobiles as first class, or even primary, devices.

4 Informal Paradigm: Technology Outside the School

A precise definition for informal learning is hard to pin down, as informal learning is often described in terms of how it differs from traditional schools. Within this paper, informal learning refers to education where students have no interactive human supervision and engage with learning materials based on their own initiative. The informal paradigm is attractive in some ways. School and teacher barriers are sidestepped, learning only student barriers. Computer-based informal learning was not previously a possibility in the developing world, but the spread of web-capable mobiles is changing this drastically. However, while informal learning offers strong appeal, it is likely a case of "the grass is always greener on the other side." First, while students' basic ICT skills were not a major problem in other contexts, studies have found that even setting up mobile Internet on phones can be an onerous task in the developing world (Gitau et al., 2010). So then, users probably need help from community centers or user groups to get started. Independent access to ICT is also required: students need a working phone or laptop with Internet capability. Informal usage also removes the constantly-available classroom peer group, limiting collaborative work and technical help. While students may naturally form study groups, the frequency and effectiveness of such emergent groups needs further study.

However, more so than any other factor, student motivation is an imposing barrier to the success of informal learning. In a traditional classroom, students can either do their work, sit idly, or incur punishment for performing off-task behavior. By comparison, informal learning environments compete with the Internet. Students need a high motivation toward the learning content to focus on an educational technology without a societal framework. No combination of teacher or school barriers may be more formidable than competing on a level playing field against the combined forces of the online media market. A pure informal paradigm may be an uphill battle. Informal learning technologies need find or create ecological niches that learners find useful and interesting.

One way to do this is to dominate a small ecological niche. One example of this is to preload devices with educational software. This paradigm relies on users trying out default programs first, rather than installing other programs. In less-developed areas, data may also be expensive enough to impose this barrier. A multi-week study on unsupervised use of preloaded educational games on mobile phones in India found that participants averaged of 2 hours and 23 minutes per week on the game, with 46 total hours per participant on average (Kumar et al., 2010). Possibly due to the game-based delivery, students had a fairly high level of motivation to learn. While off-task use was also present (e.g., downloading music), social dominance was a larger issue. Girls were particularly vulnerable, with brothers taking their phone and parents condoning this behavior. Additionally, software must be loaded onto devices at some point, so government or industry partnerships would be required for this to scale.

A second approach is to enhance existing niches, such as informal paradigms built around emergent communities of practice. For example, Mobile-ED offered a mobile gateway to a Wiki site where users could text a term and hear the web page read to them (Ford and Leinonen, 2009). If a term did not exist, users could dictate a definition that other users could use. Integrating web communities, which tend to be based on interests, and community organizations, which tend to be based on local ties, might drive sustainable informal learning, particularly on practical subject matter such as health, economic, or vocational competencies. Community groups can provide local motivational and technical peer support, as well as form connections with other user groups. By serving the shared needs of community groups, informal systems might benefit from grassroots support.

5 Conclusions and Future Directions

Classroom, institutional, and informal paradigms can each play a valuable role in developing world education immediately and in the future. While access to ICT is expanding, most of the developing world still has little computing hardware available. With that said, in raw numbers, the developing world has a strong demand for educational software that fits its needs. Primary and secondary school classrooms can benefit from shared computing applications today, through multiple-input laptops. In the future, single-display groupware or shared voice-input mobile devices might offer cheaper and equally effective designs. To make this jump, research on user interfaces for shared computing and group learning is essential. As technology evolves, regular research on these topics will be pivotal for keeping up with shifts in access and usage patterns.

Similarly, universities can immediately benefit from ubiquitous systems focused strongly on mobile learning. In the future, ubiquitous systems should be available at earlier grade levels as mobile computing expands and data costs fall. However, creating content for inexpensive mobile learning is non-trivial and many existing open systems, such as MIT Open Courseware (Abelson, 2008), are not well-suited for low resource contexts as they rely on rich media (e.g., streaming lectures). Research on methods to quickly convert existing content designed for high-resource computers (e.g., monitors, high bandwidth) to low-resource mobiles (e.g., small screen, speakers, low bandwidth) would be valuable. Techniques for rapid language and cultural localization may also be essential.

Finally, the role of informal learning in the developing world is still taking shape. Informal learning systems must target ecological niches created by technological and societal influences. While sustained engagement has been observed, social biases and gender barriers are reproduced in informal learning contexts (Kumar et al., 2010). Game-based learning and systems designed for community groups are two areas that may offer traction for supporting self-regulated education. Research on peer support and sustaining motivation for informal learning is essential, so that informal learning is both effective and equitable.

References

Abelson, H.: The creation of opencourseware at MIT. Journal of Science Education and Technology 17(2), 164–174 (May 2008)

Alcoholado, C., Nussbaum, M., Tagle, A., Gomez, F., Denardin, F., Susaeta, H., Villalta, M., Toyama, K.: One mouse per child: Interpersonal computer for individual arithmetic practice. Journal of Computer Assisted Learning 28(4), 295–309 (2012)

Bingimlas, K.: Barriers to the successful integration of ICT in teaching and learning environments: A review of the literature. Eurasia Journal of Mathematics, Science and Technology Education 5(3), 235–245 (2009)

Brunskill, E., Garg, S., Tseng, C., Findlater, L.: Evaluating an adaptive multi-user educational tool for low-resource environments. In: ICTD 2010. London, UK (2010)

Cabada, R.Z., Estrada, M.L.B., Parra, L.E., Garcia, C.A.R.: Interpreter for the deployment of intelligent tutoring systems in mobile devices. In: ICALT 2011. pp. 339–340. IEEE Press, Washington, DC (2011)

Ford, M., Leinonen, T.: Mobile-ED: Mobile tools and services platform for formal and informal learning. In: Ally, M. (ed.) Mobile Learning: Transforming the Delivery of Education and Training, pp. 195–214. AU Press (2009)

Gitau, S., Marsden, G., Donner, J.: After access: Challenges facing mobile-only internet users in the developing world. In: SIGCHI 2010. pp. 2603–06. ACM Press, New York, NY (2010)

Gulati, S.: Technology-enhanced learning in developing nations: A review. International Review of Research in Open and Distance Learning 9(1), 1–16 (2008)

International Telecommunication Union: Measuring the Information Society. Geneva, Switzerland (2012)

Kumar, A., Reddy, P., Tewari, A., Agrawal, R., Kam, M.: Improving literacy in developing countries using speech recognition-supported games on mobile devices. In: SIGCHI 2012. pp. 1149–1158. ACM Press (2012)

Kumar, A., Tewari, A., Shroff, G., Chittamuru, D., Kam, M., Canny, J.: An exploratory study of unsupervised mobile learning in rural india. In: SIGCHI 2010. pp. 743–752. ACM Press, New York, NY (Apr 2010)

Lowther, D.L., Inan, F.A., Strahl, J.D., Ross, S.M.: Does technology integration "work" when key barriers are removed? Educational Media International 45(3), 195–213 (2008)

Moreno, K.N., Klettke, B., Nibbaragandla, K., Graesser, A.C.: Perceived characteristics and pedagogical efficacy of animated conversational agents. In: Cerri, S.A., Gouardères, G., Paraguaçu, F. (eds.) Intelligent Tutoring Systems (ITS) 2002. LNCS, vol. 2363, pp. 963–971. Springer, Berlin (Jun 2002)

Munoz-Organero, M., Munoz-Merino, P.J., Kloos, C.D.: Sending learning pills to mobile devices in class to enhance student performance and motivation in network services configuration courses. IEEE Transactions on Education 55(1), 83–87 (2012)

Ogan, A., Walker, E., Baker, R.S., Rebolledo Mendez, G., Jimenez Castro, M., Laurentino, T., de Carvalho, A.: Collaboration in cognitive tutor use in Latin America: Field study and design recommendations. In: SIGCHI 2012. pp. 1381–1390. ACM Press, New York, NY (2012)

Patra, R., Pal, J., Nedevschi, S., Plauche, M., Pawar, U.: Usage models of classroom computing in developing regions. In: ICTD 2007. pp. 158–167. IEEE (Dec 2007)

Woolf, B.P., Arroyo, I., Zualkernan, I.A.: Education technology for the developing world. In: 2011 IEEE Global Humanitarian Technology Conference. pp. 493–498. IEEE (Oct 2011)

Zualkernan, I.A.: InfoCoral: Open-source hardware for low-cost, high-density concurrent simple response ubiquitous systems. In: ICALT 2011. pp. 638–639. IEEE Press (2011)

Applying Standards and AI to Educational Technology: The IEEE Actionable Data Book Project

Avron Barr, LETSI Foundation

Tyde Richards, IEEE Learning Technology Standards Committee

Dr. Robby Robson, Eduworks Corporation

1 Introduction

In the United States and other developed countries, new products built on emerging technologies such as tablets, mobile devices, cloud-based services and eBooks have generated widespread discussion about disruptive change in education at all levels. Typical questions raised include:

- Should the classroom be flipped using online video [1]?
- Can textbooks be replaced by open educational resources [2, 3]?
- What can children learn online on their own, and how can their families help?
- Can student advancement in school be tied to competence instead of cohort?
- Can a professor effectively teach 10,000 students at once in a MOOC [4-6]?
- Are automated assessments as good as human teachers [7-9]?

Although significant change is now occurring in the United States, especially in higher education, the potential for change and innovation may be even greater in the developing world. As has been demonstrated in mobile and Internet technologies, countries with less advanced infrastructures and fewer established policies and institutions can leapfrog the West in both quality of service and speed of deployment. In addition, developing countries have requirements and constraints that can lead to disruptive innovations that would not be developed in the West. An example of this may be found in the history of radio [10]. As the story goes, post-world war II Germany was given a very limited portion of the regulated radio spectrum. They therefore started using unregulated high frequencies. AM did not work well there, so they used FM. It turned out that FM had superior sound quality and became the dominant technology for quality radio.

In this context a central question is: *How can developed and developing countries collaborate to take advantage of the strengths of each?* In this paper we argue that fruitful collaboration can take place in the area of standardization and give a concrete example of how requirements from Bali spurred innovation and how standards activities in the area of eBooks may provide solutions. But first we examine in general terms the technological and related standards landscape that is emerging in eLearning.

2 Changes in eLearning Infrastructure

Today, commercial eLearning sales in the United States are dominated by two product categories, "content" (e.g. course packs and supplements to textbooks) and learning management systems (LMS). According to the Campus Computing Survey, about half of higher education institutions used an LMS in 2007 [11]. By 2011 not only did virtually all universities use an LMS [12], but only 7% had not standardized on a *single* institutional LMS [13]. From an institutional, teacher and student perspective the LMS is responsible for:

- Managing student credentials and class rosters
- Tracking entitlements to publisher content that is delivered by the LMS
- Recording student activity, task completion, and assessment results
- Analyzing and reporting results for the purposes grades and institutional research
- Delivering content and managing online communication with students
- Grading (via online assessments) and reporting grades

Most of these functions save time and money. Teachers like the LMS because it alleviates the tedium of grading, students like the "anywhere, anytime" access, administrators like them because they provide data and visibility, and publishers like the LMS because it provides a method to distribute, control and monetize their digitized intellectual property. As a result, the educational technology ecosystem found in higher education today is highly LMS-centric [14]. In recent years, many K-12 schools and jurisdictions have also invested in LMS technology. Other common educational technologies, including authoring tools, learning content management systems, assessment engines, and repositories, have been heavily influenced by the need to produce content that can be delivered via an LMS. In other words, the LMS is the dominant channel for formal learning, much as television once was for video [15].

This state of affairs has been changing for several years now as newer types of learning content have become more prevalent, including mobile apps, video lectures, online meetings, social learning, eBooks, games, and simulations. The typical LMS course contains didactic content and quizzes with pre-determined answers (e.g. multiple choice, matching and fill-in-the-blank questions), whereas these newer types of content tend to be more interactive and open ended in their assessment if student outcomes. User management and tracking results are still important in formal educational settings and for publishers' business models, but app stores and sites like YouTube are more natural delivery platforms for mobile and video content. "Learning content" is being replaced by "learning applications" that are hosted as mobile apps or as web applications in the cloud. Moreover, many of the most widely used and freely available courses (MOOCS) generate their own certificates of completion and are by their nature not tied to any one institution and therefore not to any institution's LMS.

3 Emerging Standards

As a consequence of these changes, the technical standards used by eLearning systems are being updated and revised to enable distributed systems to securely exchange data across the web [16]. This trend includes the IMS Global Learning Consortium's *Learning Tools Interoperability* (LTI) and *Learning Information Services* (LIS) specifications [17, 18] and the *Experience API* (also known as "Tin Can") produced by the U.S. Advanced Distributed Learning (ADL) initiative [19]. These standards enable applications to communicate without a central broker such as an LMS. They support interoperable reporting of assessment outcomes, course completions, and additional data relevant to learning experiences.

The capabilities offered by these emerging standards are critical for the adoption of the next generation of learning applications. For example, products such as ALEKS [20, 21], Autotutor [22, 23], Brainrush [24], Carnegie Learning [21, 25], Knewton [26], Wyang Outpost [16], and many others [27] are using embedded AI and, in some cases, game dynamics to create more effective and more engaging learning experiences. Students are now using these resources (and others such as the Kahn Academy and MOOCS) because they are either more effective or more available than traditional educational offerings. However, for these products to gain market acceptance they must be able to integrate with the ambient eLearning infrastructure. At some point schools, parents, and employers will want to see *evidence* of achievement. These systems will need to communicate results to institutional LMSs, online data repositories, and a variety of personal management apps running on the mobile devices of students, teachers, and parents.

4 New Product Categories

In addition to intelligent learning applications, many other new product categories are likely to emerge. Some will be engendered by societal requirements and others by advances in educational technology.

For example, students and teachers are increasingly associated with multiple institutions at the same time [28], and many of the more innovative learning technologies (including MOOCS and most of the systems listed earlier) are typically used outside standard classroom practice. This leads to requirements to track rosters, assignments, progress, and grades across multiple institutions and multiple online learning systems and to maintain a student's preferences in a "learner model" [29-31] that can be updated and exchanged by multiple adaptive learning systems. The natural evolution of the e-portfolio will be a personal learning record store that:

- Is securely controlled by the learner;
- Is portable as the learner works with multiple schools, teachers, tutors, and publishers over the years; and

- Contains the learner's preferences and his validated and certified formal and informal learning history.

This evolution would parallel the recent evolution of Electronic Health Records and, if implemented on a global scale, would spawn a plethora of products, ranging from tools to manage learning records to learning applications that take advantage of them to deliver more personalize, culturally relevant, and educational effective learning experiences.

Similarly, advances in cognitive science, computer science and information technology are also creating both requirements and affordances for new product categories. Just as the underlying technological components of expert systems have now found their way into hundreds of products from rice cookers to mobile phones, we anticipate that the AI components of today's intelligent tutoring systems will work their way into a wide range of learning products. The same is true for automated language understanding [32], automated grading [33], affect detection [34, 35], gesture and sketch recognition [36-38], and forms of social media that enable students to collaborate with each other and with adults (e.g. "granny tutors") [39].

Returning to the theme of standards, we observe that as learning products incorporate more intelligent features, they will generate and require significantly more data about learners, learning activities, and outcomes. Their commercial success will depend in part on their ability to create value by leveraging these data across multiple systems, jurisdictions, and stages of a life. Economically, it makes sense for learning systems to share their data rather than to hoard it, which is why standardized formats for data exchange are so important.

5 The IEEE Actionable Data Book Project

As pointed out above, standards help learning technologies integrate with existing infrastructure and processes. This means that innovations developed to meet the needs of a niche market – say one dominated by relatively low bandwidth cellular access, or one in which a culture demands different levels and types of privacy – can be used in other markets as well. Tools originally created for broader (or wealthier) markets would be more easily tailored for use elsewhere. As real-world example of a project where standards, new technologies, and unique requirements from a developing country have converged, we examine the IEEE *Actionable Data Book Project for STEM Education*, or more simply the IEEE ADB project [40].

The IEEE ADB project grew out of paper presented at the IEEE Global Humanitarian Technology Conference in 2011 that discussed a broadly applicable framework for building educational applications that combined field data collection and data visualization [41]. The requirements for the system presented in that paper came from the rice ecosystem management on the Indonesian island of Bali. In 2013, the suggestions in the paper were actualized in the IEEE ADB project. The goal of this one-year

R&D collaboration is to define and demonstrate an "actionable data book" consisting of a specialized eBook based on open standards that is tailored to support STEM education and supports learner accessibility and usage preferences. The requirements for the actionable data book are that it must be able to

- Use camera and GPS data from a learner's mobile platform
- Use measurements from local lab equipment
- Exchange results of learning interactions with cloud-based LMSs, analytics engines, and other applications
- Retrieve content from cloud-based sources (e.g. content repositories)
- Store and retrieve student history and preferences in the cloud

Operationally, the project is hosted by Industry Connections, an IEEE Standards Association program that facilitates the early exploration of potential interoperability solutions [42]. Participation is free and open to interested parties. The ADB project may continue past the initial year's charter, depending upon success.

Technologically, the project anticipates the global availability of a class of mobile devices comprising smart phones and connected tablets and explores the premise that those devices, in conjunction with a new content format, may provide the first truly global platform for connected learning. The format in question is EPUB 3 [43, 44], a new eBook format defined by the International Digital Publishing Forum [45].

EBooks have emerged as a mass-market commercial success within the past few years. To date, eBooks have only replicated the static content of printed books in a digital medium, but EPUB 3 introduces interactivity to eBooks by embracing JavaScript and the HTML5 standard for web page content. These characteristics make EPUB 3 an attractive foundation for a learning delivery platform. EPUB 3 offers a complete solution for portable, interactive, connected content, and it is relatively simple to map the requirements for an interactive learning activity onto baseline EPUB 3 capabilities. Since EPUB 3 is a general-purpose technology with broad appeal outside of the education industry, it is more likely than education-specific standards to be widely adopted, supported, and have a multi-decade life span.

Although most of the technology used by the IEEE ADB project was developed for commercial purposes in the developed world, its application to learning was originally inspired by the desire to enable students in remote locations to collect field data and share their data and culture with students in the United States. The first use case to which it will be applied is the construction of an enhanced, interactive guidebook for the new UNESCO World Heritage site on Bali [46-48].

The UNESCO site covers a significant geographical area encompassing 21 communities engaged in rice production and following traditional spiritual practices. This has resulted in an enormous challenge: How does one design an interactive guidebook that promotes the conservation and preservation of the site while meeting the needs of the people who live there, the international team developing and maintaining the site, and tourists from all over the world with varying degrees of cultural sensitivity? The IEEE ADB project aims to help meet these requirements by developing onsite learning activities and guides that adapt to the local geography and culture as well as to those of the user's culture, while also providing remote connectivity to that allows students to vicariously experience the site from anywhere on the planet.

6 Conclusions

In developing economies new policies, institutions, and business models will transform the way education is delivered and managed. These efforts will take advantage of a wide range of innovative educational technologies and products to create local solutions that overcome geographical, social, and economic barriers using global infrastructure. It is easy to envision detailed student background information being securely available via the Internet and learning systems that compete with each other on the basis of how effectively they use this information.

Similarly, as more opportunities become available for students to access online video, daily lectures may become a thing of the past and expensive, classroom-based instruction may be needed less frequently or used differently, e.g. only for activities that require in-person group interactions or that use equipment not available in homes. Independent, trusted assessment services [49, 50] may allow students to progress in school based on their acquired competence, displacing today's cohort-based advancement schemes that measure progress by seat-time. The possibilities are unlimited and each educational jurisdiction will shape their solution by their specific needs and resources.

Data exchange standards and software interoperability standards are key to the flexible configuration of future systems, online services, and mobile applications. Standards-based products allow a school or a national or regional education agency to configure multiple products, including their current systems, into a stable working solution that fits local requirements and that allows new capabilities to be incorporated over time with minimal effort. The IEEE Actionable Data Book project is an example of a new model for learning delivery based on globally available, open standards that focuses on the realities of teaching and learning in the developing world.

7 REFERENCES

1. Khan, S., *The One World Schoolhouse: Education Reimagined*. 2012: Twelve.
2. Hylén, J., D.V. Damme, and F. Mulder, *Open Educational Resources*. 2012.
3. Wiley, D., et al., *A preliminary examination of the cost savings and learning impacts of using open textbooks in middle and high school science classes*. The International Review of Research in Open and Distance Learning, 2012. **13**(3): p. 262-276.

4. Anderson, T., *Promise and/or Peril: MOOCs and Open and Distance Education*. 2013.
5. Belanger, Y. and J. Thornton, *Bioelectricity: A Quantitative Approach Duke University's First MOOC*. 2013.
6. Ruth, S., *Can MOOC's and Existing E-Learning Efficiency Paradigms Help Reduce College Costs?* Available at SSRN 2086689, 2012.
7. Murray, K.W. and N. Orii, *Automatic Essay Scoring*.
8. Tsai, M.-h., *The Consistency Between Human Raters and an Automated Essay Scoring System in Grading High School Students' English Writing*. Action in Teacher Education, 2012. **34**(4): p. 328-335.
9. Vujosevic-Janicic, M., et al., *Software Verification and Graph Similarity for Automated Evaluation of Students' Assignments*. arXiv preprint arXiv:1206.7064, 2012.
10. Wikipedia. *History of Radio*. 2013 [cited 2013 May 17]; Available from: https://en.wikipedia.org/wiki/History_of_radio.
11. Green, C., *CAMPUS COMPUTING, 2007*, 2007, Claremont, CA: Claremont Graduate University.
12. Dahlstrom, E., et al., *with a foreword by Diana Oblinger*. The ECAR National Study of Undergraduate Students and Information Technology, 2011 (Research Report), 2011.
13. Green, C., *CAMPUS COMPUTING, 2011*, 2011, campuscomput.net.
14. Mott, J., *Envisioning the post-LMS era: the Open Learning Network*. Educause Quarterly, 2010. **33**(1): p. 1-9.
15. Frieden, R., *Next Generation Television and the Migration from Channels to Platforms*. Available at SSRN 2117960, 2012.
16. Alario-Hoyos, C. and S. Wilson. *Comparison of the main alternatives to the integration of external tools in different platforms*. in *Proceedings of the International Conference of Education, Research and Innovation, ICERI*. 2010. Citeseer.
17. GLC, I. *Learning Information Services*. 2012 [cited 2013 May 17]; Available from: http://www.imsglobal.org/lis/.
18. GLC, I. *Learning Tools Interoperability*. 2012 [cited 2013 May 17]; Available from: http://www.imsglobal.org/toolsinteroperability2.cfm.
19. ADL. *Training & Learning Architecture (TLA): Experience API (xAPI)*. 2013 [cited 2013 May 17]; Available from: http://www.adlnet.gov/tla/experience-api.
20. ALEKS. *ALEKS Home Page*. 2011 [cited 2011 October 20]; Available from: http://www.aleks.com.
21. Sabo, K.E., et al., *Searching for the two sigma advantage: Evaluating algebra intelligent tutors*. Computers in Human Behavior, 2013. **29**(4): p. 1833-1840.
22. Graesser, A., et al., *AutoTutor: An Intelligent Tutoring System with Mixed-Initiative Dialogue*, in *IEEE Transactions on Education*2005, IEEE. p. 612-618.
23. Hu, X., et al. *AutoTutor Lite*. 2009. IOS Press.
24. Brainrush. *Brainrush Home Page*. 2013 [cited 2013 May 17]; Available from: http://www.brainrush.com/.
25. Carnegie Learning. *Carnegie Learning Home Page*. 2013 [cited 2013 May 17]; Available from: http://www.carnegielearning.com/.
26. Knewton. *Knewton improves learning outcomes (Knewton Home Page)*. 2013 [cited 2013 May 17]; Available from: http://www.knewton.com/.

27. Institute of Education Sciences. *What Works Clearinghouse*. 2013 [cited 2013 May 17]; Available from: http://ies.ed.gov/ncee/wwc/.
28. Newbaker, P., *No Straight Path to College Graduation*, 2012.
29. Sottilare, R., et al., *Design Recommendations for Adaptive Intelligent Tutoring Systems: Learner Modeling (Vol. 1)*. 2013: Army Research Lab.
30. Woolf, B.P., *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. 2009, Burlington, MA: Morgan Kaufmann.
31. Durlach, P.J. and J.M. Ray, *Designing adaptive instructional environments: Insights from empirical evidence*, in *Army Research Institute Report*2011: Arlington, VA.
32. Robson, R. and F. Ray. *Applying Semantic Analysis to Training, Education, and Immersive Learning*. in *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*. 2012. NTSA.
33. Valenti, S., F. Neri, and A. Cucchiarelli, *An overview of current research on automated essay grading*. Journal of Information Technology Education, 2003. **2**: p. 319-330.
34. Hussain, M., et al. *Affect detection from multichannel physiology during learning sessions with AutoTutor*. in *Artificial Intelligence in Education*. 2011. Springer.
35. Calvo, R.A. and S. D'Mello, *Affect detection: An interdisciplinary review of models, methods, and their applications*. Affective Computing, IEEE Transactions on, 2010. **1**(1): p. 18-37.
36. Weinland, D., R. Ronfard, and E. Boyer, *A survey of vision-based methods for action representation, segmentation and recognition*. Computer Vision and Image Understanding, 2011. **115**(2): p. 224-241.
37. Yin, P., et al. *Sketch worksheets: A sketch-based educational software system*. in *Proceedings of the 22nd Innovative Applications of Artificial Intelligence Conference, Atlanta*. 2010.
38. Valentine, S., et al., *Mechanix: A Sketch-Based Tutoring and Grading System for Free-Body Diagrams*. AI Magazine, 2012. **34**(1): p. 55.
39. Doctorow, C. *Deploying the British Granny Cloud to tutor poor Indian classrooms over Skype*. 2011 [cited 2013 May 17]; Available from: http://boingboing.net/2011/01/20/deploying-the-britis.html.
40. IEEE ADB Project. *Public Home Page: The IEEE Actionable Data Book for S.T.E.M. Education*. 2013 [cited 2013 May 17]; Available from: http://www.ieee-adb.org/.
41. Richards, T. and A. Barr. *Catalyzing Connected Learning through Standards*. in *Global Humanitarian Technology Conference (GHTC), 2011 IEEE*. 2011. IEEE.
42. Richards, T. *IEEE Actionable Data Book for STEM Education: Industry Connections Activity Initiation Document (ICAID)*. 2012 [cited 2013 May 17]; Available from: http://standards.ieee.org/about/sasb/iccom/IC12-006-02_IEEE_Actionable_Data_Book_for_STEM_Education.pdf.
43. Garrish, M., *What is EPUB 3?* 2011: O'Reilly Media.
44. IDPF. *EPUB 3*. 2013 [cited 2013 May 17]; Available from: http://idpf.org/epub/30.
45. IDPF. *Home Page*. 2013 [cited 2013 May 17]; Available from: http://idpf.org/.
46. Lansing, J.S. and J.N. Watson, *Guide to Bali's UNESCO World Heritage, Tri Hita Karana: Cultural Landscape of Subaks and Water Temples*, 2012.

47. UNESCO. *Cultural Landscape of Bali Province: the Subak System as a Manifestation of the Tri Hita Karana Philosophy*. 2012 [cited 2013 May 17]; Available from: http://whc.unesco.org/en/list/1194/.
48. Lansing, J.S. and J.N. Watson. *Water temples forever (Video)*. 2012 [cited 2013 May 17]; Available from: http://www.youtube.com/watch?v=jPetj4MSDdY.
49. Morgan, J. and A. Millin, *ONLINE PROCTORING PROCESS FOR DISTANCE-BASED TESTING*, 2011, Google Patents.
50. Kitahara, R., F. Westfall, and J. Mankelwicz, *New, multi-faceted hybrid approaches to ensuring academic integrity*. Journal of Academic and Business Ethics, 2011. **3**: p. 1-12.

AIED 2013 Workshops Proceedings
Volume 7

Recommendations for Authoring, Instructional Strategies and Analysis for Intelligent Tutoring Systems (ITS): Towards the Development of a Generalized Intelligent Framework for Tutoring (GIFT)

Workshop Co-Chairs:

Robert A. Sottilare & Heather K. Holden
Army Research Laboratory, Human Research and Engineering Directorate, Orlando, Florida, USA

https://gifttutoring.org/news/14

Preface

This workshop provides the AIED community with an in-depth exploration of the Army Research Laboratory's effort to develop tools, methods and standards for Intelligent Tutoring Systems (ITS) as part of their Generalized Intelligent Framework for Tutoring (GIFT) research project. GIFT is a modular, service-oriented architecture developed to address authoring, instructional strategies, and analysis constraints currently limiting the use and reuse of ITS today. Such constraints include high development costs; lack of standards; and inadequate adaptability to support tailored needs of the learner. GIFT's three primary objectives are to provide: (1) authoring tools for developing new ITS, ITS components (e.g., learner models, pedagogical models, user interfaces, sensor interfaces), tools, and methods based on authoring standards that support reuse and leverage external training environments; (2) an instructional manager that encompasses best tutoring principles, strategies, and tactics for use in ITS; and (3) an experimental testbed for analyzing the effect of ITS components, tools, and methods. GIFT is based on a learner-centric approach with the goal of improving linkages in the adaptive tutoring learning effect chain in Figure 1.

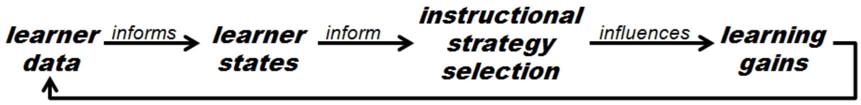


Figure 1: Adaptive Tutoring Learning Effect Chain

The goal of GIFT is to make ITS affordable, effective, usable by the masses, and provide equivalent (or better) instruction than expert human tutors in one-to-one and one-to-many educational and training domains. GIFT's modular design and standard messaging provides a largely domain-independent approach to tutoring where domain-dependent information is concentrated in the one module making most of its components, tools and methods reusable across training domains. More information about GIFT can be found at www.GIFTtutoring.org.

The workshop is divided into five themes: **(1)** *Fundamentals of GIFT* (includes a tutorial on GIFT and a detailed demonstration of the latest release); **(2)** *Authoring ITS using the GIFT Authoring Construct*; **(3)** *Adapting Instructional Strategies and Tactics using GIFT*; **(4)** *Analyzing Effect using GIFT*; and **(5)** *Learner Modeling*. Themes include presentations from GIFT users regarding their experiences within the respective areas and their recommendations of design enhancements for future GIFT releases. Theme 5 is dedicated to discussing the outcomes of the learner modeling advisory board meeting conducted at the University of Memphis Meeting in September 2012.

July, 2013
Robert Sottilare, Heather Holden.

Program Committee

Co-Chair: Robert Sottilare, Army Research Laboratory, Orlando, FL, USA
(robert.a.sottilare@us.army.mil)

Co-Chair: Heather Holden, Army Research Laboratory, Orlando, FL, USA
(heather.k.holden@us.army.mil)

Arthur Graesser, *University of Memphis*

Xiangen Hu, *University of Memphis*

James Lester, *North Carolina State University*

Ryan Baker, *Columbia University*

Table of Contents

Motivations for a Generalized Intelligent Framework for Tutoring (GIFT) for Authoring, Instruction and Analysis

Robert A. Sottilare, Ph.D. and Heather K. Holden, Ph.D.

U.S. Army Research Laboratory – Human Research and Engineering Directorate
{robert.sottilare, heather.k.holden}@us.army.mil

Abstract. Intelligent Tutoring Systems (ITS) have been shown to be effective tools for one-to-one tutoring in a variety of well-defined domains (e.g., mathematics, physics) and offer distinct advantages over traditional classroom teaching/training. In examining the barriers to the widespread use of ITS, the time and cost for designing and author-ing ITS have been widely cited as the primary obstacles. Contributing factors to time and cost include a lack of standards and minimal opportunities for reuse. This paper explores motivations for the development of a Generalized Intelligent Framework for Tutoring (GIFT). GIFT was conceived to meet challenges to: author ITS and ITS components, offer best instructional practices across a variety of training tasks (e.g., cognitive, affective, and psychomotor), and provide a testbed for analyzing the effect of tutoring technologies (tools and methods).

1 Introduction

GIFT [1] is a modular, service-oriented architecture developed to address authoring, instructional strategies, and analysis constraints currently limiting the use and reuse of ITS today. Such constraints include high development costs; lack of standards; and inadequate adaptability to support tailored needs of the learner. GIFT's three primary objectives are to develop: (1) authoring tools to develop new ITS, ITS components (e.g., learner models, pedagogical models, user interfaces, sensor interfaces), tools, and methods, and develop authoring standards to support reuse and leveraging external training environments; (2) provide an instructional manager that encompasses best tutoring principles, strategies, and tactics for use in ITS; and (3) an experimental testbed to analyze the effect of ITS components, tools, and methods. GIFT is based on a learner-centric approach with the goal of improving linkages in the adaptive tutoring learning effect chain in Figure 1.

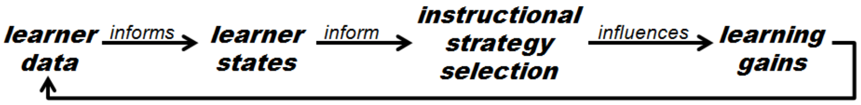


Figure 1: Adaptive Tutoring Learning Effect Chain [2]

GIFT's modular design and standard messaging provides a largely domain-independent approach to tutoring where domain-dependent information is concentrated in the domain module making most of its components, tools and methods reusable across tutoring scenarios.

2 Motivations for authoring tools, standards and best practices

The primary goal of GIFT is to make ITS affordable, usable by the masses, and equivalent (or better) than an expert human tutors in one-to-one and one-to-many educational and training scenarios for both well-defined and ill-defined domains. As ITS seek to become more adaptive to provide tailored tutoring experiences for each learner, the amount of content (e.g., interactive multimedia and feedback) required to support additional adaptive learning paths grows exponentially. More authoring requirements generally means longer development timelines and increased development costs. If ITS are to be ubiquitous, affordable, and holistically learner-centric, it is essential to for ITS designers and developers to develop methods to rapidly author content or reuse existing content. Overcoming barriers to reuse means developing standards. In this context, the idea for GIFT was born.

2.1 GIFT Authoring Goals

Adapted from Murray [3] [4] and Sottilare and Gilbert [5], the authoring goals discussed below identify several motivating factors for the development of authoring methods and standards. First and foremost, the idea of a GIFT is founded on decreasing the effort (time, cost, and/or other resources) required to author and analyze the effect of ITS, ITS components, instructional methods, learner models, and domain content. ITS must become affordable and easy to build so we should strive to decrease the skill threshold by tailoring tools for specific disciplines to author, analyze and employ ITS.

In this context, we should provide tools to aid designers, authors, trainers/teachers, and researchers organize their knowledge for retrieval and application at a later time. Automation should be used to the maximum extent possible to data mine rich repositories of information to create expert models, misconception libraries, and hierarchical path plans for course concepts.

A GIFT should support (structure, recommend, or enforce) good design principles in its pedagogy, its user interface, etc. It should enable rapid prototyping of ITS to allow for rapid design/evaluation cycles of prototype capabilities. To support reuse, a GIFT should employ standards to support rapid integration of external training/tutoring environments (e.g., serious games) to leverage their engaging context and avoid authoring altogether.

2.2 Serious Games and ITS

Serious games, which are computer-based games aimed at training and education rather than pure entertainment, are one option for reuse if they can easily be integrated with tutoring architectures like GIFT. Serious games offer high-level interactive multi-media instructional (IMI) content that is engaging and is capable of supporting a variety of scenarios with the same basic content. While most serious games offer prescriptive feedback based on learner task performance, the integration of serious games with ITS opens up the possibility of more adaptive feedback based on a more comprehensive learner model.

In order to facilitate the use of serious games in a tutoring context (game-based tutoring), standards are needed to support the linkage of game actions to learning objectives in the tutor. To this end, Sottilare and Gilbert [5] recommend the development of two standard interface layers, one layer for the game and one for the tutor. The game interface layer captures entity state data (e.g., behavioral data represented in the game), game state data (physical environment data), and interaction data, and passes this information to the tutor interface layer. The tutor interface layer passes data from the game to the instructional engine which develops strategies and tactics (e.g., feedback and scenario changes) which are passed back to the game to initiate actions (e.g., non-player character provides feedback or challenge level of scenario is increased).

Additional options for reuse should be explored to minimize/eliminate the amount of authoring required by ITS designers and developers. The ability to structure approaches for configuring a variety of tutoring experiences and experiments is discussed next.

2.3 Configuring tutoring experiences and experiments

Another element of authoring is the ability to easily configure the sequence of instruction by reusing standard components in a script. This is accomplished in GIFT though a set of XML configuration tools used to sequence tutoring and/or experiments. Standard tools include, but are not limited to functional user modeling, learner modeling, sensor configuration, domain knowledge file authoring, and survey authoring which are discussed below.

While not yet implemented in GIFT, functional user models are standard structures and graphical user interfaces used to facilitate tasks and access to information that is specific to the type of user (e.g., learners, subject matter experts, instructional system designers, system developers, trainers/instructors/teachers, and scientists/researchers).

Learner models are a subset of function user models used to define what the ITS needs to know about the learner in order to inform sound pedagogical decisions per the adaptive tutoring learning effect model. The learner configuration authoring tool provides a simple tree structure driven by XML schema which prevents learner model authoring errors by validating inputs against the learner model XML schema. This configuration tool also provides ability to validate the learner model using GIFT source without having to launch the entire GIFT architecture. Inputs to the learner modeling configuration include translators, classifiers, and clustering methods which use learner data to inform learner states (e.g., cognitive and affective).

The sensor configuration authoring tool allows the user to determine which sensors will be used during a given session and which translators, classifiers, and clustering methods the sensor data will feed. Again, this is an XML-based tool which allows the user to select a combination of behavioral and physiological sensor to support their tutoring session or experiment. Several commercial sensors have been integrated into GIFT through plug-ins.

Survey authoring is accomplished through the GIFT survey authoring system (SAS) which allows the generation and retrieval of questions in various formats (e.g., true/false, multiple choice, Likert scales) to support assessments and surveys to support tailoring decisions within GIFT. Through this tool, questions can be associated with assessments/surveys and these in turn can be associated with a specific tutoring event or experiment.

Domain authoring is accomplished through the domain knowledge file authoring tool. This tool allows an instructional designer to sequence events (e.g., scenarios, surveys, content presentation). GIFT currently support various tutoring environments expand the flexibility of course construction. These include Microsoft PowerPoint for content presentation, surveys and assessments from the GIFT SAS, serious games (e.g., VMedic and Virtual BattleSpace (VBS) 2). More environments are needed to support the variety of tasks that might be trained using GIFT.

3 Motivations for expert instruction

Significant research has been conducted to model expert human tutors and to apply these models to ITS to make them more adaptive to the needs of the learner without the intervention of a human instructor. The INSPIRE model [6] [7] is noteworthy based on the extensive scope of this studies that led to this model. Person and others [8] [9] seek to compare and contrast how human tutors and ITS might most effectively tailor tutoring experiences.

For its initial instructional model a strategy-tactic ontology, the *engine for Macro-Adaptive Pedagogy* (eMAP), was developed based on Merrill's Component Display Theory [10], the literature, and variables that included the type of task (e.g., cognitive, affective) and instruction (e.g., individual, small group instruction). Instructional strategies are defined as domain-independent policies that are implemented by the pedagogical engine based on input about the learner's state (e.g., cognitive, affective, domain-independent progress assessment (at expectation, below expectation, or above expectation)). Strategies are recommendations to the domain module in GIFT which selects a domain-dependent tactic (action) based on the strategy type (e.g., prompt, hint, question, remediation) and specific instructional context, where the learner is in the instructional content.

A goal for GIFT is for it to be a nexus for capturing best practices from tutoring research in a single place where scientists can compare the learning effect of each model and then evolve new models based on the best attributes of each model analyzed. To support this evolution, GIFT includes a testbed methodology called the *analysis construct* which is discussed below.

4 Motivations for an effect analysis testbed

As noted in the previous section, GIFT includes an analysis construct which is not only intended to evolve the development of expert instructional models, but is also available to analyze other aspects of ITS including learner modeling, expert modeling, and domain modeling. The notion of a GIFT analysis construct shown in Figure 2 was adapted from Hanks, Pollack, and Cohen's testbed methodology [11].

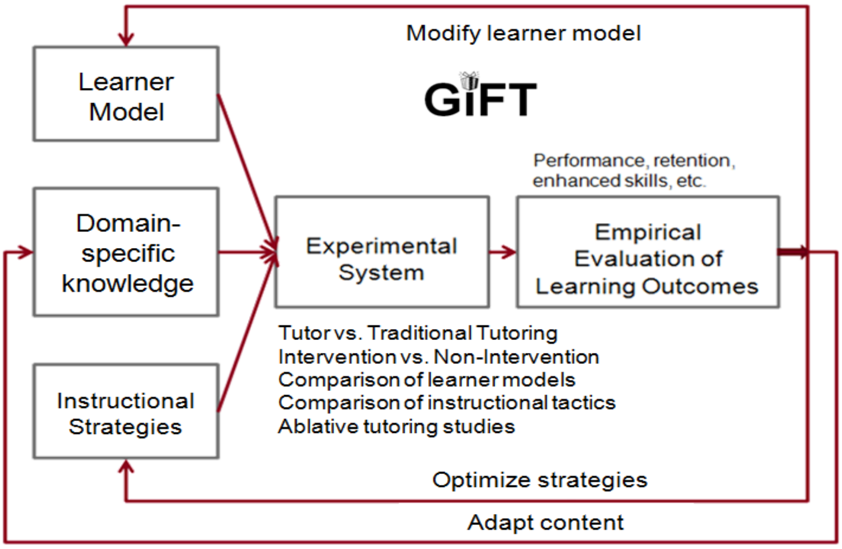

Modify learner model
GIFT
Learner Model
Domain-specific knowledge
Instructional Strategies
Experimental System
Performance, retention, enhanced skills, etc.
Empirical Evaluation of Learning Outcomes
Tutor vs. Traditional Tutoring
Intervention vs. Non-Intervention
Comparison of learner models
Comparison of instructional tactics
Ablative tutoring studies
Optimize strategies
Adapt content

Figure 2: GIFT analysis construct

A great benefit of GIFT's analysis construct it is ability to conduct comparisons of whole tutoring systems as well as specific components (either entire models or specific model elements). To date, ITS research has been limited in its ability to conduct such comparative analyses due to the high costs associated with redesign and experimentation. This construct can be leveraged to assess the impact and interplay of both learner characteristics directly contributing to the learning process (i.e., abilities, cognition, affect, learning preferences, etc.) and those that are external and indirectly effect the learning process (i.e., perceptions of technology, the ITS interface, and learning with technology, etc.). Similarly, GIFT can provide formative and summative assessments to identify the influence of various instructional strategies and tactics; based on these assessments, GIFT is able to better improve and guide instruction dynamically and more effectively.

Across all levels of education and training populations, regardless of the mode of instruction (i.e., live, virtual, or constructive), a paradigm shift in the learning process is occurring due to the evolution of technology and the increase in ubiquitous computing. This notion has become noticeably apparent over the last few years. Even Bloom's revised taxonomy has been recently updated to account for new actions, behaviors, processes, and learning opportunities brought forth by web-based technology advancements [12]. Moreover, with the increasing recognition of the importance of individual learning differences in instruction, GIFT can ultimately be able to support the educational framework and principles of the universal design for learning (UDL) [13, 14]. This framework highlights the need for multiple means of r*epresen-*

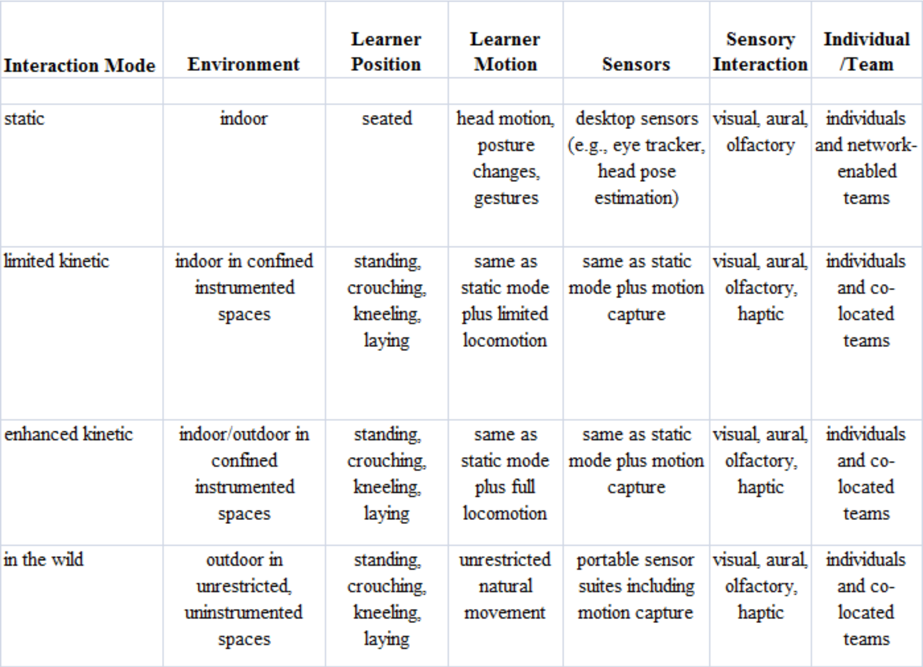
tation, *expression*, and *engagement* to reduce barriers of learning and provide fruitful learning experiences for all types of learners. While this concept has evolved over the past decade, practicality and experimentation to progress this notion to true reality has been limited. However, GIFT's analysis construct can be used to access the effectiveness of UDL principles in an empirically-driven fashion.

5 Expanding the horizons of ITS through future GIFT capabilities

The potential of GIFT is dependent on two primary objectives: 1) focus research and best practices into authoring, instructional, and analysis tools and methods within GIFT to enhance its value to the ITS community and 2) expanding the horizons of traditional ITS outside the bounds of traditional ITS. This section concentrates on examining areas for future development which will expand the current state-of-practice for ITS including tutoring domains, interaction modes, and automation processes for authoring.

The application of ITS technologies has largely been limited to one-to-one, well-defined tutoring domains where information, concepts, and problems are presented to the learner and the learner's response is expected to correspond to a single correct answer. This works well for mathematics, physics and other procedurally-driven domains (e.g., first aid), but not as well for ill-defined domains (e.g., exercises in moral judgment) where there might be more than one correct answer and these answers vary only by their level of effectiveness. It should be a goal of the ITS community to develop an ontology for use developing and analyzing tutors for ill-defined domains.

Traditional tutors have also been generally limited to static interaction modes where a single learner is seated in front of a computer workstation and interaction is through a keyboard, mouse, or voice interface. Methods to increase the learner's interaction and range of motion are needed to move ITS from cognitive and affective domains to psychomotor and social interaction domains. It should be a goal of the ITS community to develop additional interaction modes to support increasingly natural training environments for both individuals and teams as shown in Table 1.



Interaction Mode	Environment	Learner Position	Learner Motion	Sensors	Sensory Interaction	Individual /Team
static	indoor	seated	head motion, posture changes, gestures	desktop sensors (e.g., eye tracker, head pose estimation)	visual, aural, olfactory	individuals and network-enabled teams
limited kinetic	indoor in confined instrumented spaces	standing, crouching, kneeling, laying	same as static mode plus limited locomotion	same as static mode plus motion capture	visual, aural, olfactory, haptic	individuals and co-located teams
enhanced kinetic	indoor/outdoor in confined instrumented spaces	standing, crouching, kneeling, laying	same as static mode plus full locomotion	same as static mode plus motion capture	visual, aural, olfactory, haptic	individuals and co-located teams
in the wild	outdoor in unrestricted, uninstrumented spaces	standing, crouching, kneeling, laying	unrestricted natural movement	portable sensor suites including motion capture	visual, aural, olfactory, haptic	individuals and co-located teams

Table 1. ITS interaction modes

Automation processes should be developed to support authoring of expert models, domain models, and classification models for various learner states (cognitive, affective, and physical). Data mining techniques should be optimized to define not only expert performance, but also levels of proficiency and expectations based on a persistent (long-term) learner model. Again, data mining techniques are needed to reduce the time and cost to author domain models including automated path planning for courses based on the hierarchical relationship of concepts, the development of misconception libraries based on course profiles, feedback libraries (e.g., questions, prompts) based on readily available documentation on the internet and from other sources .

6 References

1. Sottilare, R.A., Brawner, K.W., Goldberg, B.S., & and Holden, H.K. (2012). The Generalized Intelligent Framework for Tutoring (GIFT). Orlando, FL: U.S. Army Research Laboratory – Human Research & Engineering Directorate (ARL-HRED).
2. Sottilare, R. (2012). Considerations in the development of an ontology for a Generalized Intelligent Framework for Tutoring. *International Defense & Homeland Security Simulation Workshop in Proceedings of the I3M Conference*. Vienna, Austria, September 2012.
3. Murray, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education*, 10(1):98–129.

4. Murray, T. (2003). An Overview of Intelligent Tutoring System Authoring Tools: Updated analysis of the state of the art. *Authoring tools for advanced technology learning environments*. 2003, 491-545.
5. Sottilare, R., & and Gilbert, S. (2011). Considerations for tutoring, cognitive modeling, authoring and interaction design in serious games. *Authoring Simulation and Game-based Intelligent Tutoring workshop at the Artificial Intelligence in Education Conference (AIED) 2011*, Auckland, New Zealand, June 2011.
6. Lepper, M. R., Drake, M., & O'Donnell-Johnson, T. M. (1997). Scaffolding techniques of expert human tutors. In K. Hogan & M. Pressley (Eds), *Scaffolding student learning: Instructional approaches and issues* (pp. 108-144). New York: Brookline Books.
7. Lepper, M. and Woolverton, M. (2002). The Wisdom of Practice: Lessons Learned from the Study of Highly Effective Tutors. In J. Aronson (Ed.), Improving academic achievement: impact of psychological factors on education (pp. 135-158). New York: Academic Press.
8. Person, N. K., & Graesser, A. C., & The Tutoring Research Group (2003). Fourteen facts about human tutoring: Food for thought for ITS developers. Artificial Intelligence in Education 2003 Workshop Proceedings on *Tutorial Dialogue Systems: With a View Toward the Classroom* (pp. 335-344). Sydney, Australia.
9. Person, N. K., Kreuz, R. J., Zwaan, R. A., & Graesser, A. C. (1995). Pragmatics and pedagogy: Conversational rules and politeness strategies may inhibit effective tutoring. *Cognition and Instruction*, 13(2), 161–188.
10. Merrill, M. D. (1983). Component Display Theory. In C. M. Reigeluth (Ed). *Instructional-design theories and models: An overview of their current status*, pp.279-333. Hillsdale, NJ: Lawrence Erlbaum Associates.
11. Hanks, S., Pollack, M.E. and Cohen, P.R. (1993). Benchmarks, Test Beds, Controlled Experimentation, and the Design of Agent Architectures. *AI Magazine*, Volume 14 Number 4.
12. Churches, A. 2009. Retrieved April 29, 2013 from http://edorigami.wikispaces.com/Bloom's+Digital+Taxonomy.
13. King-Sears, M. (2009). Universal Design for Learning: Technology and Pedagogy, *Learning Disability Quarterly*, 32(4), 199-201.
14. Rose, D.H. and Meyer, Anne. (2002). Teaching Every Student in the Digital Age: Universal Design for Learning. Association for Supervision and Curriculum Development, ISBN-0-87120-599-8.

7 Acknowledgment

Research was sponsored and conducted by the Army Research Laboratory. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. Addition information about GIFT can be found at **www.GIFTtutoring.org**.

Authors

Robert A. Sottilare, PhD serves as the Chief Technology Officer (CTO) of the Simulation & Training Technology Center (STTC) within the Army Research Laboratory's Human Research and Engineering Directorate (ARL-HRED). He also leads adaptive tutoring research within ARL's Learning in Intelligent Tutoring Environments (LITE) Laboratory where the focus of his research is in automated authoring, instructional management, and analysis tools and methods for intelligent tutoring systems. His work is widely published and includes recent articles in the *Cognitive Technology* and the *Educational Technology* Journals. Dr. Sottilare is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT). He received his doctorate in Modeling & Simulation from the University of Central Florida with a focus in intelligent systems. In January 2012, he was honored as the inaugural recipient of the U.S. Army Research Development & Engineering Command's Modeling & Simulation Lifetime Achievement Award.

Heather K. Holden, Ph.D. is a researcher in the Learning in Intelligent Tutoring Environments (LITE) Lab within the U.S. Army Research Laboratory – Human Research and Engineering Directorate (ARL-HRED). The focus of her research is in artificial intelligence and its application to education and training; technology acceptance and Human-Computer Interaction. Dr. Holden's doctoral research evaluated the relationship between teachers' technology acceptance and usage behaviors to better understand the perceived usability and utilization of job-related technologies. Her work has been published in the Journal of Research on Technology in Education, the International Journal of Mobile Learning and Organization, the Interactive Technology and Smart Education Journal, and several relevant conference proceedings. Her PhD and MS were earned in Information Systems from the University of Maryland, Baltimore County. Dr. Holden also possesses a BS in Computer Science from the University of Maryland, Eastern Shore.

Unwrapping GIFT

A Primer on Developing with the Generalized Intelligent Framework for Tutoring

Charles Ragusa, Michael Hoffman, and Jon Leonard

Dignitas Technologies, LLC, Orlando, Florida, USA
{cragusa,mhoffman,jleonard}@dignitastechnologies.com

Abstract. The Generalized Intelligent Framework for Tutoring (GIFT) is an open-source, modular, service-oriented framework which provides tools, methods and services designed to augment third-party training applications for the purpose of creating intelligent and adaptive tutoring systems. In this paper we provide a high-level overview of GIFT from the technical perspective, and describe the key tasks required to integrate a new training application. The paper will be most helpful for software developers using GIFT, but may also be of interest to instructional designers, and others involved in course development.

Keywords: Adaptive Tutoring, Intelligent Tutoring, Framework, Pedagogy

1 Introduction

The Generalized Intelligent Framework for Tutoring (GIFT) is a framework and tool set for the creation of intelligent and adaptive tutoring systems[1-3]. In its current form GIFT is largely an R&D tool designed to provide a flexible experimentation platform for researchers in the intelligent and adaptive tutoring field. However, as GIFT matures, it moves ever closer to becoming a production quality framework suitable for use in fielded training systems.

Generally speaking, GIFT is domain and training application agnostic. And, while it can present generic content such as documents, multi-media content, etc.; specialized content is typically presented via an external software system, which we will refer to as a training application (TA). GIFT provides a standardized way to integrate training applications and includes many tools and services required to transform the TA into an intelligent and/or adaptive tutoring system. Services and standards include:

- Standard approach for interfacing training applications
- Domain knowledge representation (including authoring tool)
- Performance assessment
- Course flow (including authoring tool)
- Pedagogical model including micro and macro adaptation
- Learner modeling

- Survey support (with authoring tools)
- Learning management system
- Standardized approach for integrating physiological (and other) sensors

Another key aspect of GIFT is that it is an open source project[1]. Baseline development is currently performed by Dignitas Technologies; however, where appropriate, community developed capabilities will be rolled back into the baseline. In addition, results from current and upcoming experiments, such as pedagogical models, learner models, etc. may eventually be incorporated into future releases. Thus, GIFT is an evolving and ever-improving system, where individual contributions are re-integrated into the baseline for the mutual benefit of all users in the community.

2 Architecture

The GIFT runtime environment uses a service-oriented architecture and consists of several loosely coupled modules, communicating via asynchronous message passing across a shared message bus. Key modules and their primary functions are:

- **Gateway Module**: Connects GIFT to third-party training applications.
- **Sensor Module**: Connects GIFT to physiological sensors in a standardized way.
- **Learner Module**: Models the cognitive, affective, and performance state of the learner [4].
- **Pedagogical Module**: Responsible for making domain-independent pedagogical decisions, using an internal pedagogical model based on learner state.
- **Domain Module**: Performs performance assessment, based on domain-expert authored rules, carries out domain specific implementations of pedagogical actions based on domain-independent pedagogical requests, and (together with the pedagogical module) orchestrates course flow.
- **Tutor Module**: Presents the user interface for tasks such as presentation of surveys, providing feedback, engaging in two-way dialogues, etc.
- **Learning Management System (LMS) Module**: GIFT connection to an external learning management system, for the storage and maintenance of learner records, biographical data, course material etc.
- **User Management System (UMS) Module**: Manages users of the GIFT system, manages surveys and survey data, and provides logging functions.
- **Monitor Module**: Non-critical module, used as control panel for starting and stopping other GIFT modules, and monitoring the state of active GIFT sessions.

[1] GIFT users are encouraged to register on the GIFT portal at http://gifttutoring.org. The site provides access to the latest builds, source code, documentation, and supports active forums for general discussion and trouble-shooting.

3 Getting Started with the GIFT Framework

3.1 GIFT Messages

Message Classes. GIFT messages are the sole means of communication between GIFT modules. The Message class hierarchy consists of three classes. The Message base class includes all boiler-plate message fields such as the time stamp, the payload type, an object reference for the optional payload, identification of the source and destination modules, etc. Two subclasses add additional fields appropriate for the GIFT context, such as User Session ID and Domain Session ID.

Message Payloads. Many message types transport data in the optional payload. To support inter-process communication (IPC), the messages and their payloads must comply with an agreed upon encoding and decoding scheme. In GIFT 3.0 the default scheme is Java Script Object Notation (JSON).

Message Types. Every GIFT message has an associated type. The various message types are enumerated in the class mil.arl.gift.common.enums.MessageTypeEnum.

3.2 Interfacing a Training Application using the GIFT Gateway Module

Training Application Considerations. There are two basic requirements that a TA must meet for a satisfactory integration with GIFT. The first is a means to transmit game state from the TA to GIFT. The second is a way for GIFT to exercise some degree of control over the TA. Basic controls such as launching the TA, loading specific content, and shutting down the TA, are very helpful in making a seamless training solution, even though they are not strictly required.

The requirement to communicate game state is immediately met if the TA includes a facility for communicating via a standardized network protocol such as Distributed Interactive Simulation (DIS) protocol. In the absence of such a capability, the TA must be augmented either by leveraging an existing API or by modifying the TA's source code to allow communication of the game state to GIFT via IPC.

Control of the TA by GIFT follows a similar pattern. If an existing protocol exists, it should be used. If not, then custom development will be required. In addition to basic start, load, stop-type control messages, some use cases may require more advanced interactions, discussion of which is beyond the scope of this document.

Creating the Gateway Module Plugin. The process of adapting a TA to the GIFT gateway module involves creating a gateway module plugin. When faced with integrating a new TA, a developer should first ask if one of the existing plugins is suitable for reuse. GIFT 3.0 includes plugins for: DIS, Power Point, TC3Sim, and VBS2. Even if a new plugin is required, these will serve as excellent references.

When developing a new plugin, the primary objective is to implement a concrete subclass of mil.arl.gift.gateway.interop.AbstractInteropInterface. The essential requirements of a new subclass are minimal, but by providing concrete implementations for each of the abstract methods, the plugin will seamlessly operate within the gateway module context. Beyond that, the plugin should implement whatever additional functionality it requires, such as receiving game state messages from the TA and converting them to GIFT messages, and/or receiving GIFT messages (e.g. SIMAN messages) and passing them on to the TA in a way that the TA will understand.

GIFT Messaging. To complete the integration of the TA with the gateway module, at least one GIFT message payload class is needed to represent the game state of the TA. Existing message payload classes that have been used with previously integrated TA's include: TC3GameStateJSON, EntityStateJSON, and PowerPointJSON. If any of these satisfactorily represents the game state from the new TA, then reusing the existing message is advised. However, if none of them are suitable, then a new message will be required. Any new message payload types should be added to the mil.arl.gift.common.enums.MessageTypeEnum class and the appropriate payload class(es) added to the mil.arl.gift.net.api.message.codec.json package.

3.3 Domain Module Modifications and Programming

Overview. At the appropriate time(s) during the execution of a GIFT course, the domain module loads a domain-specific file called the domain knowledge file (DKF). This XML input file contains the domain specific information required by the domain module to carry out several of its key tasks during the learner's interaction with the TA. The first is assessments of the learner's performance on various training tasks encountered during the TA session. It also includes micro-pedagogical mappings of learner state (affective, cognitive, and performance) transitions to named instructional strategies as well as implementation details of those strategies**.**

Integration of any new TA, or even developing a new training course using a previously integrated TA, will typically require DKF authoring as a primary task. In some cases, new custom java coding may also be required, as discussed below.

Domain Knowledge File Authoring. Given that DKF files are XML, they can be edited with any number of text or XML editors, but the preferred method is to use the GIFT-supplied DKF authoring tool (DAT). Using the DAT will enforce the DKF schema as well as perform other validation such as checks against external references.

Before creating a new DKF the user should become familiar with the DKF file format, which is described in the file GIFTDomainKnowledgeFile.htm[2]. In addition, a GIFT release may include one or more test documents (spreadsheets), one of which will contain a step-by-step procedure for authoring a DKF from scratch.

Performance Assessment Authoring. Performance assessment authoring is done within the assessment tag of the DKF file. The basic structure is a task/concept/condition

[2] This and many other documents are contained in the GIFT/docs folder within the GIFT source distribution, which is available for download at http://gifttutoring.org

hierarchy. Tasks have start and end triggers and a set of concepts. Each concept, in turn, will have a set of conditions[3]. It is at the condition level that computation takes place. In fact, you'll notice that each condition tag will contain a conditionImpl tag that refers to a java class responsible for carrying out the performance computation based upon game state received from the TA and inputs encoded in the DKF. Currently, performance values are limited to: unknown, below expectation, at expectation, and above expectation. Beyond the runtime performance assessment, each condition also supports a set of authorable scoring rules and evaluators that together determine the final score for that condition. When scoring rules are present, learners are presented with an after-action review of their performance at appropriate times and scores are written to the LMS.

State Transition Authoring. State transition authoring is performed within the actions tag of the DKF. The basic structure is a list of state transitions, each of which represent a state change in the learner, to which the tutor should react, along with a list of strategy choices (options) that may be used when that particular state change is encountered. In cases where state transitions refer to the learner's performance state, the state transition will have a reference back to a performance node in the assessment section of the DKF.

Instructional Strategy Authoring. Instructional strategy (IS) authoring is also performed within the actions tag of the DKF. Implemented strategies currently include learner feedback, scenario adaptations (changes to the currently executing TA scenario), and request for performance assessment by the domain module. Each strategy entry references a StrategyHandler, which is a specification of the java class responsible for handling authored input contained in the DKF file. The linkage to java code allows substantial flexibility as will be discussed in the next section.

Custom Programming. As described above, the domain module supports a built-in scheme for extending its capabilities for both performance assessment and for instructional strategies. To augment the performance assessment capabilities, a developer codes an implementation of the AbstractCondition interface and then references the implementation class in the appropriate section of the DKF. The key abstract method to be implemented is the handleSimulationMessage[4] method, which takes in a Message as the sole argument, and returns an AssessmentLevelEnum. The message argument is, of course, a representation of the game state that originates in the TA. Developers of new condition implementations should strive to make their code as abstract as possible to allow for the broadest possible reuse[5].

[3] This is a simplified description for the sake of readability. In actuality, concepts support arbitrarily deep nesting of other concepts (i.e., sub-concepts).

[4] The method name reflects GIFT's early development focus on integration with simulations such as VBS2. In future releases the name will be likely be changed to something more generic, such as, "handleGameStateMessage".

[5] Reuse across different TA's, scenarios, domains, etc.

Implementing new instructional strategies is done similarly. Developers provide a concrete implementation of the StrategyHandlerInterface and then reference the implementation class within the DKF. A good example of this is seen with providing feedback to a learner. In the DefaultStrategyHandler, feedback is presented to the learner using the GIFT Tutor User Interface (TUI). However, in a recent experiment, alternative presentations of feedback were required. To satisfy this requirement the TC3StrategyHandler was developed, which allowed feedback strings to be communicated back to TC3Sim for presentation to the learner directly by TC3Sim.

3.4 Surveys and Survey Authoring

GIFT uses the term "survey" to refer generically to any number of interrogative forms presented to the learner via the TUI. GIFT supports survey authoring through its Survey Authoring System (SAS) web application as well as runtime presentation and processing of surveys during execution of a GIFT course. GIFT surveys can be used for a variety of purposes including pre, mid, and post lesson competency assessment; acquiring biographical and demographic information; psychological and learner profiling; and even for user satisfaction surveys. A variety of useful question and response types are supported. Further discussion of the SAS is beyond the scope of this document, but interested readers can consult the GIFTSurveyAuthoringSystemInstructions.htm for additional information.

3.5 Course Authoring

Currently in GIFT, the top-level unit of instruction that learners interact with is a called a course, the specification of which is contained in a dedicated course.xml file. Prior to GIFT 3.0 a course specified a fixed linear flow through a series of course elements; however, with GIFT 3.0 we have introduced support for dynamic flow through course elements, based on macroadaptation strategies.

The primary course elements are surveys, lesson material, and TA sessions. Survey elements administer GIFT surveys that have been previously authored using the SAS. Lesson material elements present browser compatible instructional content such as PDF documents, html pages, or other media files. TA sessions support interactive sessions with a TA such as VBS2, PowerPoint, or other specialized software systems. A fourth course element called "Guidance", which presents textual messages to the learner, exists to support making user-friendly transitions between other course elements. For example at the conclusion of a survey a guidance element might be used to introduce an upcoming TA session.

Course.xml files are authored using the Course Authoring Tool (CAT). For linear flow, the author uses the CAT to specify the various elements of the course along with any necessary inputs. For dynamic flow, authoring involves selecting when in the course flow a branch point is appropriate. The branch point specifies that the macro pedagogical model should gather a list of metadata attributes based on the current learner state when deemed necessary. This collection of metadata attributes is then provided to the domain module as search criteria over the domain content resources for the current course. As the search discovers domain content matching the metadata attributes of interest, paradata files are used to drill down the list of possible content to display based on usage data. The end result is that the domain module is able to present content based on the learner state and pedagogy recommendations.

3.6 Learner Module

The Learner Module is responsible for managing learner state, which can include short term and predicted measures of cognitive and affective state as well as other long term traits. Inputs used to compute state can originate from multiple sources including TA performance assessments sent from the domain module, sensor data from the sensor module, survey responses, and long term traits stored in the LMS.

To date, the GIFT team has focused on computing learner state from sensor data received from the sensor module. The processing framework employs a pipeline architecture which allows the developer to chain concrete implementations of abstract data translators, classifiers, and predictors. Customized pipelines can be created for each sensor type and/or groups of sensors.

Creation of pipelines using existing java implementation classes is performed using the Learner Configuration Authoring Tool, which is launched using scripts/launchLCAT.bat. Currently defined pipelines can be found in GIFT/config/learner/LearnerConfiguration.xml.

Developers requiring customized implementation classes are referred to the API docs and source code in the mil.arl.gift.learner package. Key abstract classes include AbstractClassifier, AbstractBasePredictor, and AbstractSensorTranslator.

Measurement, representation, and application of learner state are areas of active research and future version of the GIFT learner module will incorporate relevant research outcomes to enhance its capabilities.

3.7 Pedagogical Module

The pedagogical module is responsible for making pedagogical decisions based on learner performance and state. Its primary objective is to reason on the available information, and then influence the training environment to maximize the learning effectiveness for each individual learner using the system. The rules, algorithms, and heuristics that provide the basis for making pedagogical decisions in a domain-independent way are generally referred to as the pedagogical model. One near term goal of GIFT is to provide a framework upon which intelligent tutoring researchers can easily integrate, test and validate a variety of pedagogical models.

In GIFT 3.0, there are two pedagogical models in place: a micro and a macro model. The micro model uses the state transitions information authored in the DKFs as described in previous sections. The macro model is based on research gathered by IST on macro adaptive pedagogy findings [5] which has been encoded as an XML file in GIFT. This XML file is used to configure the macro adaptive pedagogical model when the pedagogical module is started. The information contains a tree-like structure specifying useful metadata for different types of learner state characteristics. This model will continue to be developed after GIFT 3.0.

3.8 Learning Management System (LMS) Module

The GIFT LMS module is a surrogate for an external LMS. In the future, a commercial grade LMS system may be integrated to maintain a variety of data, including student records, course material, and other learning resources. However, in the current version of GIFT, the LMS implementation is an SQL database, designed simply to store and maintain learner records for GIFT courses that have been completed. Aside from developers engaged in integration of GIFT with a production LMS system, very few developers will have a need to modify the LMS module.

3.9 User Management System (UMS) Module

The UMS module is supports three major functions: management of users; storage and maintenance of the surveys, survey questions and learner responses to surveys; and message logging. None of these functions are likely targets for development for new GIFT users; however, the logging feature is very important for researchers.

The UMS-managed log files contain every message sent between the various modules during each GIFT session. Using the GIFT Event Reporting Tool (ERT) researchers can apply filters to the log files to isolate messages of interest and perform analysis and data mining that can be used to construct new models.

3.10 Tutor Module: User Interface Considerations

Users interact with GIFT via the TUI, which is a web application that connects to GIFT on the back end. As of GIFT 3.0, Internet Explorer 9.0 is the browser of choice, in accordance with the current U.S. Army mandate [6]. Learner interactions with the TUI include: user login, surveys, feedback, after-action review, interactive dialogues, learning material presentation, etc.

3.11 Monitor Module

As of GIFT 3.0 the monitor module is largely a tool used to launch various GIFT modules and serve as a monitor of a running GIFT session. It is an unlikely development target for new GIFT users. Use of the Monitor Module is described in GIFTMonitor(IOS-EOS)Instructions.htm.

3.12 Sensor Module and Sensor Configuration

The Sensor Module provides a standardized approach to acquiring data from sensors measuring some aspect of Learner State. Currently integrated sensors include: EEG (Emotiv), Electro Dermal Activity (QSensor), Palm temperature and humidity (via instrumented mouse), Zephyr-Technology BioHarness, Inertial Labs Weapon Orientation Module (WOM), USC/ICT Multisense, and Microsoft Kinect.

Sensor data are sent to the learner module to become part of the learner state and potentially used by the pedagogical module. Time-stamped sensor data are also written to log files making them available for post-run analysis by researchers.

The sensor module is configured pre-runtime by editing the SensorConfig.xml file using the Sensor Configuration Authoring Tool (SCAT). The SensorConfig.xml file specifies which sensors should be activated by the sensor module, which plugin (java class) to load to access the sensor hardware, as well as any specialized configuration data. In addition, the SensorConfig.xml includes specification of Filters and Writers, which control the filtering of raw sensor data and writing of sensor data to log files. Users can specify which sensor configuration file is used by editing the GIFT/config/sensor/sensor.properties file.

Developers using one of the previously integrated sensors can, in most cases, limit their focus to editing of the SensorConfig.xml file using the SCAT. Developers integrating new sensors will need to write java code. The key coding task for required for creating a new sensor plugin is to implement a concrete subclass of AbstractSensor. Developers may also want to subclass AbstractSensorFilter and/or AbstractFileWriter, though there are default implementations of these classes that will suffice for many applications.

4 Conclusion

GIFT is a highly configurable and extensible open-source framework designed to support a wide range of intelligent and adaptive tutoring applications. Its modularity and configurability make it well suited for a variety of research efforts.

Configuration and customization opportunities are available at a number of levels ranging from minor editing of text-based configuration files to creation of new java classes. Basic module settings are configurable in dedicated java properties files located in GIFT/config subfolders. More sophisticated configurations reside in XML files, which, depending on the purpose, may reside in a GIFT/config subfolder (e.g. SensorConfig.xml) or alongside the domain content (e.g., course.xml and dkf.xml). GIFT includes specialized editors/authoring tools for many of these files.

As an open-source project, users also have the ability to extend GIFT by modifying source code. In key areas where user extensions are anticipated, GIFT uses appropriate object oriented abstractions. Developers are then able to create their own customized implementation classes, and specify their use at runtime by edits made to the corresponding XML file.

Interested parties are encouraged to register on the GIFT Portal at (http://gifttutoring.org).

5 References

1. Sottilare, R. A., Brawner, K. W., Goldberg, B. S., & Holden, H. K. (2012). The Generalized Intelligent Framework for Tutoring (GIFT).
2. Brawner, K., Holden, H., Goldberg, B., & Sottilare, R. (2012). Recommendations for Modern Tools to Author Tutoring Systems. In *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*(Vol. 2012, No. 1). National Training Systems Association.
3. Sottilare, R. A., Goldberg, B. S., Brawner, K. W., & Holden, H. K. (2012). A Modular Framework to Support the Authoring and Assessment of Adaptive Computer-Based Tutor-

ing Systems (CBTS). In *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*(Vol. 2012, No. 1). National Training Systems Association.

4. Holden, H. K., Sottilare, R. A., Goldberg, B. S., & Brawner, K. W. (2012). Effective Learner Modeling for Computer-Based Tutoring of Cognitive and Affective Tasks. In *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)* (Vol. 2012, No. 1). National Training Systems Association.
5. Goldberg, B., Brawner, K., Sottilare, R., Tarr, R., Billings, D. R., & Malone, N. (2012). Use of Evidence-based Strategies to Enhance the Extensibility of Adaptive Tutoring Technologies. In *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)* (Vol. 2012, No. 1). National Training Systems Association.
6. Information Assurance Vulnerability Alert 2012-A-0198 ARMY IAVA - Revised-2013-A-5001 {Release Jan 11 2013}.

Authors

Charles Ragusa is a senior software engineer at Dignitas Technologies with over thirteen years of software development experience. After graduating from University of Central Florida with a B.S. in computer science, Mr. Ragusa spent several years at SAIC working on a variety of R&D projects in roles ranging from software engineer and technical/integration lead to project manager. Noteworthy projects include the 2006 DARPA Grand Challenge as an embedded engineer with the Carnegie Mellon Red Team, program manager of the SAIC CDT/MRAP IR&D project, and lead engineer for Psychosocial Performance Factors in Space Dwelling Groups. Since joining Dignitas Technologies in 2009, he has held technical leadership roles on multiple projects, including his current role as the principal investigator for the GIFT project.

Michael Hoffman is a software engineer at Dignitas Technologies with over seven years of experience in software development. Upon graduating from the University of Central Florida with a B.S. in Computer Science, he spent a majority of his time on various OneSAF development activities for SAIC. He worked on the DARPA Urban Challenge, where he provided a training environment for the robot by simulating AI traffic and the various sensors located on Georgia Tech's Porsche Cayenne in a OneSAF environment. Soon after earning a Master of Science degree from the University of Central Florida, Michael found himself working at Dignitas. Both at Dignitas and on his own time, Michael has created several iPhone applications. One application, called the Tactical Terrain Analysis app, provides mobile situation awareness and can be used as a training tool for various real world scenarios. More recently he has worked to determine if unobtrusive sensors can be used to detect an individual's mood during a series of computer interactions. Michael excels in integrating both software and hardware systems such as third party simulations and sensors. Michael has been the lead software engineer on GIFT since its inception over two years ago.

Jon Leonard is a junior software engineer at Dignitas Technologies. He graduated from the University of Central Florida with a B.S. in Computer Science where he gained experience in concepts such as computer learning and computer graphics by developing an augmented reality Android video game. At Dignitas, among several other projects, he has worked on a simulation environment for M1A1 Abrams and Bradley tank gunnery training. His personal efforts include working on open source procedural content generation algorithms and mobile applications. Jon's interest is in solving interesting problems. Jon has been a developer for GIFT since its inception.

GIFT Research Transition: An Outline of Options

How transition in GIFT moves through phases of idea, project, and paper, and to real-world use.

Keith W. Brawner[1]

[1] *Army Research Laboratory*
Keith.W.Brawner@us.army.mil

Abstract. This paper describes the use of the Generalized Intelligent Framework for Tutoring (GIFT) to transition research and findings into use beyond publication. A proposal is submitted to use GIFT as a research platform for community development, with examples of how it provides transition opportunities for individual researchers. Several projects which have already transitioned are discussed, while two projects by the author are specifically shown as examples.

Keywords. Intelligent Tutoring Systems, Research Transition

1 Current Transition Path for Research in the ITS Community

The Generalized Intelligent Framework for Tutoring (GIFT) development is currently performed under contract for the Army Research Laboratory. There any many reasons why the military is interested in training technology in general, and adaptive intelligent training technologies in specific [1]. Fundamentally, the end result of research conducted at ARL is technological advancements which are usable by soldiers, or, succinctly, “Technology Driven. Warfighter Focused”.

Technology transition is defined as the process of transferring skills, knowledge, or ability from research (typically performed at university or Government labs) to users who can further develop or exploit these items into products, processes, applications, or services. There are many ways for research projects to transition from research to development, to new product, to lifecycle support. While this innovative diffusion may occur solely through technological ‘push’ of publishing or the ‘pull’ user adoption, these typically do not occur without a transition partner [2]. Part of the purpose of ARL is to function as a transition partner: leveraging technology advances made in academic laboratories, developing them into usable products, and transitioning them to developmental support roles.

ITS research has historically transitioned directly to the user, which bypasses the developmental and exploitive portions of a traditional transition. One example of this is a project such as the Cognitive Tutor, which bypassed the “external development” phase through marketing to local school districts. Another example includes the Crystal Island program, which has also transitioned through collaboration with the local school districts, rather than an industrial base. Further examples include AutoTutor transition of Operation ARIES through the facilitating intermediary of Pearson Education, or GnuTutor through open source software release.

Researchers generally face competing desires for their project. As a research goal, they desire to perform research, create findings, publish results, and solve interesting problems. A researcher may have a related goal, which competes for their time: the desire for their technology to be useful to a population of users. Given finite resources, the individual or organization must compromise one of these goals to facilitate the other. A two-way facilitating transition partner would allow the researcher to see their creation used and obtain meaningful feedback while maintaining research pursuits.

ARL in general, and the GIFT project in specific, have a goal of facilitating this research transition. This goal is not empty talk, as the repackaging and transition of several research projects has already occurred programmatically. In addition to being an ARL researcher, the author is anticipated to obtain a doctorate at the University of Central Florida in August 2013. Research done at ARL and UCF alike are both transitioning to the field through the GIFT, and will be described in this paper. The author will outline how you can use GIFT to transition your research, give examples of projects which have done so, and describe the benefits of this approach.

2 Proposal for a Community Research Platform

GIFT is intended to be both a community platform and growing standard [3, 4]. This fundamentally offers several advantages, a short selection of which is described below:

- Like any open source software approach, a researcher or developer is able to build upon the work of others. This magnifies the ability of an individual developer to contribute.
- Like any community project, a developer is able to quickly see the use of their work. An individual developer/researcher is able to quickly access a population of users of their research, which magnifies their individual impact.
- ITS technology can be leveraged against a broad amount of training content, while keeping the same core functionality. This magnifies the use of the product.
- The ITS technology can improve through various software versions, which improves learning while costing little or nothing for implementation. Content is used in a more useful fashion, making the use of an incrementally updated project attractive.
- A researcher or developer can use standardized tools to create, modify, or adjust individual items for the purpose of experimentation, evaluation, and validation.
- Experimental comparisons can be conducted fairly at multiple locations, with multiple populations. This allows the research conducted within the framework to be fairly compared.
- A researcher can leverage tools which make the interpretation of data easier. A shared set tools has been of aid to other researchers in Educational Data Mining [5].

3 Re-GIFT-ing: models of transition

There are several models of transition which can be used with varying levels of researcher interaction and levels of opportunity. Transition into GIFT may be through a tool, a compatible software or hardware product, a plug-in, a releasable item, or a piece of software integrated into an official baseline. These differing modes of transition are summarized in **Error! Reference source not found.** alongside the required user interaction, an example of a project which has followed this transition path, and the potential impact that it has to the field.

The first project to discuss is the tool created by the Personalized Assistant for Learning (PAL) for data analysis [6]. During the course of a PAL experiment with GIFT, the developers found it helpful to have a tool to parse through GIFT data. After developing this tool, they provided this it back to the community through simply posting it on the http://www.gifttutoring.org forums. An author following this transition path may host a "**GIFT tool**" on their own site, make it available to only their lab, or other method. To the author's knowledge, no one has used or modified this tool outside of their laboratory. However, others have the opportunity to use this tool and improve on it, and its functionality has directed the development of a more thorough tool available within the GIFT Release: the Event Reporting Tool (ERT).

The next project, and method of transition, to discuss is the eMotive EEG library. The eMotive EEG was found helpful in other research conducted by the author [7], and was incorporated into GIFT as a software library interface. The purchase of an eMotive EEG headset gives the developer access to the library. The fact the GIFT supports easy integration of the sensor makes it so that each GIFT user is a potential eMotive customer, which benefits eMotive. Transition of research as a "**GIFT compatible**" product involved little interaction with the developers, but may be unsupported in future releases. While developer involvement is low, the potential impact is similarly low.

Continuing to use sensors as an example, the next project to discuss is the Q-Sensor project, which transitioned in a way which is different from the previous versions. All software required to integrate an Affectiva Q-Sensor is provided freely to the GIFT community, as part of a **"GIFT plugin"**. Changes made to the Q-Sensor are supported in future versions of GIFT and the plugin is released in the current GIFT 3.0 version. To date, this type of transition has resulted in the use of Q-Sensor technology in a minimum of two different experiments, with three pilot trials. This has occurred with little interaction from the Q-Sensor developers.

There are now several complete programmed packages which are released with the GIFT version. One of these is the medical instruction and assessment game "vMedic", which contains several scenarios which have GIFT tutoring. Another example is the Human Affect Recording Tool (HART), developed by Ryan Baker [8], which enables affective coding of behavior. Both of these programs have reached a wider audience through leveraging **"GIFT releasable"** transition, with some work required by the developer. The developer of each of these programs targeted use within GIFT as part of the model of development. Each of these programs is provided back to the community as downloadable software packages on www.gifttutoring.org. In this fashion, the vMedic program has reached a significantly wider audience and the HART app has seen distribution and citation.

Lastly, one can transition source code directly into the GIFT baseline via a **“GIFT integration”**, in anticipation of the next release. The work required to integrate into the GIFT framework is done by the developer, before giving it back to www.gifttutoring.org. While this requires more work, it is able to reach a wider audience, and is automatically carried forward into each future release. This is the only release path which is thoroughly tested and vetted prior to each version. This allows for the broadest application of the developed technology.

Table 2. Examples of various GIFT transitions, projects which used this transition method, and levels of interaction provided

Type of Transition	**Example of project**	**User interaction**	**Potential Impact**
GIFT tool	PAL Tool	None	Low
GIFT compatible	eMotive EEG	None	Low
GIFT plug-in	Q-Sensor	Low	Medium
GIFT releasable	HART, vMedic	Medium	Medium
GIFT integration	GSR filtering, MultiSense	High	High

4 Two Research Transition Stories: GSR Filtering, realtime modeling

In this section, the author will tell two stories research transition where first-hand experience was obtained. The first of these stories involves the transition of a new GSR sensor filtering method, available in GIFT 2.0, while the second focuses on a larger piece of work which has intended availability in GIFT 4.0. The aim of this section is to give an example of how an idea becomes a deliverable.

4.1 GSR Filtering

The first project idea is that a realtime sensor filter may be able to collect meaningful measures of affective/cognitive state in realtime. The idea behind this project is that the author was unaware of relevant feature extraction techniques, or implementations, for several datastreams of interest. A dataset was used which collected both ECG and GSR measures while participants experienced various training events [7]. It was hypothesized that meaningful measures of cognition and affect could be extracted from these sensor datastreams.

It was found that meaningful measures of cognition and affect could be extracted, including statistical measures and signal power measures, borrowing from the field of digital signal processing. It is possible that these techniques could be leveraged into an intelligent tutoring system. These results were then published [9].

Just because a method has been published to be useful does not mean that industrial or academic partners and collaborators will take it upon themselves to read an academic paper, implement the algorithm, and put it in their system. The more that an individual developer can do to help this process, the quicker transition of the research will be [2]. One way to do this is to merge the work of a researcher with a project which is already transitioning to industry. GIFT represents this possibility.

The idea, project, and paper on GSR filtering has transitioned into GIFT via the "GIFT integration" route. Every researcher which downloads GIFT (which is compatible with a GSR sensor) is able to implement the developed feature extraction, do their own experiment and draw their own conclusions. Furthermore, any ITS constructed from the GIFT framework and tools already has this implementation, and the development of a student model which uses this information progresses a significant step towards reality. The ECG filtering from the same paper is intended to be released GIFT 4.0.

To date, GSR filtering algorithms have now been provided to over 100 researchers and developers. The author hopes that his work will be found valuable. In either case, the developed research has been placed in the hands of numerous users, which is more valuable than publication alone. If the work is not found valuable, the author would hope that the other researchers are able to improve on the technique, and feed the results to other researchers through a similar transition path.

4.2 Realtime modeling

The second project idea is that individualized models of learner affect/cognition may be able to be created in realtime. The idea behind the second project is that generalized models of affect and cognition are difficult to create. Individualized models can be made, but their quality is known to degrade over time [10]. Realtime modeling and adaptive algorithms may present a solution to the problem.

The realtime modeling project used two datasets [11] and constructed seven total classifiers. The approach used four different types of classification techniques, including neural gasses, resonance theory, clustering, and online linear regression. Each of these techniques was developed with three different schemes for labeling data, including unsupervised, semi-supervised, and fully-supervised.

It was found that semi-supervision had significant contribution to the overall accuracy of the problem. It was also found that realtime affective models could be created with reasonable quality, and that realtime cognitive models are a more difficult problem that requires alternate means in conjunction with the methods presented. These results will be published as a doctoral dissertation later in the year.

Realtime student state assessment is anticipated to be available within GIFT 4.0. Targeting GIFT as a research transition allows industry and academia to benefit from the research, and targets a larger and different audience than publication. Once again, transition of research through GIFT allows larger access, experimentation, citation, and overall exposure.

5 Conclusion/Recommendations

GIFT is a functional Intelligent Tutoring System which has been used as part of several experiments. Research which transitions into GIFT has the potential to be used by a population of learners, instructional designers, and experimenters. Each of these user groups is anticipated to have their own user interface, which can make use of the research transitioned into GIFT, in whichever fashion is implemented.

In addition, GIFT is intended as a research platform, and Army Research Laboratory has plans for development out to 2017. A research transition into GIFT, in any fashion, should be able to reach a community of users for the next four years, at a minimum. The project has potential longevity beyond 2017, with funding from the Army, DoD, or others. Even if not supported by the Army, it will remain in the public domain, able to be improved by anyone in the community. Using GIFT as an exit vector for research ideas has more potential than simple publication, or of hosting an open source project.

Furthermore, the licensing agreement on GIFT does not hinder the individual researcher from capitalizing on their ideas. Two for-profit companies have targeted GIFT as a technology which can support the ability to commercialize their ideas, while others have been in conversation. Other research organizations have proposed or used GIFT to widen their audience and to focus their expertise.

This paper has discussed how some research technology has *already* transitioned to the field using the GIFT entry vector, and how other portions are intended. The concept which the author presents in this workshop paper and presentation is that it is possible to use GIFT as a platform to transition research results into the field of use, while minimizing the effort required by the researcher.

6 References

1. Army, D.o.t., *The U.S. Army Learning Concept for 2015*, 2011: TRADOC.
2. Fowler, P. and L. Levine, *A conceptual framework for software technology transition*, 1993, DTIC Document.
3. Sottilare, R.A., et al. *A Modular Framework to Support the Authoring and Assessment of Adaptive Computer-Based Tutoring Systems (CBTS).* in *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).* 2012. NTSA.
4. Sottilare, R.A., et al., *The Generalized Intelligent Framework for Tutoring (GIFT).* 2012.
5. Koedinger, K.R., et al., *A data repository for the EDM community: The PSLC DataShop*. Handbook of Educational Data Mining, 2010: p. 43-55.
6. Regan, D., E.M. Raybourn, and P. Durlach, *Learning modeling consideration for a personalized assistant for learning (PAL)*, in *Design Recommendations for Adaptive Intelligent Tutoring Systems: Learner Modeling (Volume 1).*2013.
7. Goldberg, B.S., et al., *Predicting Learner Engagement during Well-Defined and Ill-Defined Computer-Based Intercultural Interactions*, in *Proceedings of the 4th International Conference on Affective Computing and Intelligent Interaction (ACII 2011), LNCS*, S.D. Mello, et al., Editors. 2011, Springer-Verlag: Berlin Heidelberg. p. 538-547.
8. Baker, R.S., et al. *Towards Sensor-Free Affect Detection in Cognitive Tutor Algebra.* in *Proceedings of the 5th International Conference on Educational Data Mining*. 2012.

9. Brawner, K. and B. Goldberg. *Real-Time Monitoring of ECG and GSR Signals during Computer-Based Training*. 2012. Springer.
10. AlZoubi, O., R. Calvo, and R. Stevens, *Classification of EEG for Affect Recognition: An Adaptive Approach*. AI 2009: Advances in Artificial Intelligence, 2009: p. 52-61.
11. Carroll, M., et al., *Modeling Trainee Affective and Cognitive State Using Low Cost Sensors*, in *Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*2011: Orlando, FL.

Authors

Keith W. Brawner is a researcher for the Learning in Intelligent Tutoring Environments (LITE) Lab within the U. S. Army Research Laboratory's Human Research & Engineering Directorate (ARL-HRED). He has 7 years of experience within U.S. Army and Navy acquisition, development, and research agencies. He holds a Masters degree in Computer Engineering with a focus on Intelligent Systems and Machine Learning from the University of Central Florida, and will obtain a doctoral degree in the same field in Summer 2013. The focus of his current research is in machine learning, active learning, realtime processing, datastream mining, adaptive training, affective computing, and semi/fully automated user tools for adaptive training content.

Experimentation with the Generalized Intelligent Framework for Tutoring (GIFT): A Testbed Use Case

Benjamin Goldberg[1] and Jan Cannon-Bowers[2]

[1]*United States Army Research Laboratory-Human Research & Engineering Directorate-Simulation and Training Technology Center, Orlando, FL 32826*
{benjamin.s.goldberg@us.army.mil}
[2]*Center for Advanced Medical Learning and Simulation (CAMLS) at the University of South Florida, Tampa, FL 33602*
{jcannonb@health.usf.edu}

Abstract. Computer-Based Tutoring Systems (CBTS) are grounded in instructional theory, utilizing tailored pedagogical approaches at the individual level to assist in learning and retention of associated materials. As a result, the effectiveness of such systems is dependent on the application of sound instructional tactics that take into account the strengths and weaknesses of a given learner. Researchers continue to investigate this challenge by identifying factors associated with content and guidance as it pertains to the learning process and the level of understanding an individual has for a particular domain. In terms of experimentation, a major goal is to identify specific tactics that impact an individual's performance and the information that manages their implementation. The Generalized Intelligent Framework for Tutoring (GIFT) is a valuable tool for this avenue of research, as it is a modular, domain-independent framework that enables the authoring of congruent systems that vary in terms of the research questions being addressed. This paper will present GIFT's design considerations for use as an experimental testbed, followed by the description of a use case applied to examine the modality effect of feedback during game-based training events.

Keywords: generalized intelligent framework for tutoring, instructional strategies, testbed, feedback, experimentation

1 Introduction

The overarching goal of Computer-Based Tutoring Systems (CBTSs) is to enable computer-based training applications to better serve a leaner's needs by tailoring and personalizing instruction [1]. Specifically, the goal is to achieve performance benefits within computer-based instruction as seen in Bloom's 1984 study "the 2-Sigma Problem". Though there is recent debate on the validity of these results [2], this classic experiment showed that individuals receiving one-on-one instruction with an expert tutor outperformed their fellow classmates in a traditional one-to-many condition by an average of two standard deviations. The success of this interaction is in the ability of the instructor to tailor the learning experience to the needs of the individual. Interaction is based on the knowledge level of the learner as well as their performance and reaction (i.e., cognitive and affective response) to subsequent problems and communications [3].

With the recent development of the Generalized Intelligent Framework for Tutoring (GIFT; see Figure 1), a fundamental goal is to develop a domain-independent pedagogical model that applies broad instructional strategies identified in the literature. This framework would then be used to author adaptive environments across learning tasks to produce benefits accrued through one-on-one instruction. At the core of GIFT is pedagogical modeling, which is associated with the application of learning theory based on variables empirically proven to influence performance outcomes [4]. According to Beal and Lee [5] the role of a pedagogical model is to balance the level of guidance and challenge during a learning event so as to maintain engagement and motivation. The notion for GIFT is to identify generalized strategies on both a macro- and micro-adaptive level that can be used to author specific instructional tactics for execution in a managed ITS application. The pedagogical model uses data on '*Who*' is being instructed, '*What*' is being instructed, and the '*Content*' available from which to instruct. In an ideal case, GIFT can identify recommended strategies based on this information, and also provide tools to convert those strategies into specific instructional tactics for implementation.

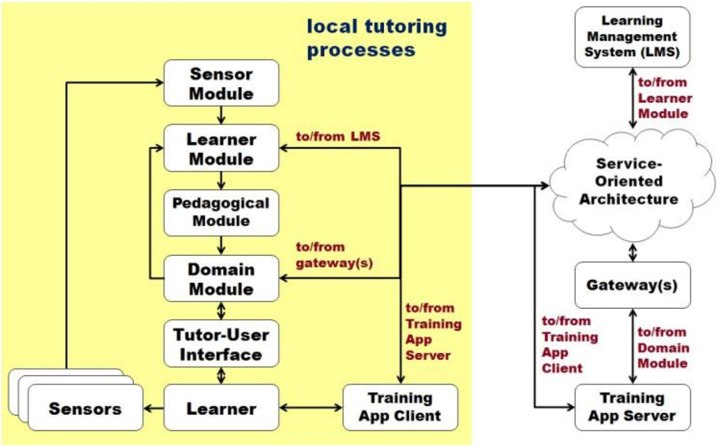


Fig. 3. Generalized Intelligent Framework for Tutoring (GIFT)

Before this conceptual approach of GIFT can be realized, a great deal work needs to be done to identify strategies found to consistently affect learning across multiple domains (codified in the pedagogical model) and the variables that influence the selection of these strategies (expressed in the learner model). In the remainder of this paper, we describe GIFT's functional application as an experimental testbed for conducting empirical research, followed by a descriptive use case of a recent instructional strategy-based experiment examining the effect varying modalities of feedback delivery have on learner performance and engagement within a game-based environment.

1.1 GIFT's Testbed Functionality

For GIFT to be effective across all facets of learning, there are a number of research questions that need to be addressed. These include, but are not limited to: (1) How can GIFT be used to manage the sequence, pace, and difficulty of instructional content before a learning session begins, as well as how to adapt instruction in real-time based on learner model metrics?; (2) What information is required in the learner model to make informed decisions on instructional strategy selection?; (3) How can GIFT best manage guidance and feedback during a learning session based on competency and individual differences?; and (4) What is the optimal approach for delivering GIFT communications to a learner during system interaction?

While GIFT provides the tools necessary to author and deliver adaptive learning applications, an additional function of the framework is to operate as a testbed for the purpose of running empirical evaluations on research questions that will influence future developmental efforts. Empirically evaluating developed models and techniques is essential to ensuring the efficacy of GIFT as a sound instructional tool. To accommodate this requirement, while maintaining domain-independency, GIFT's design is completely modular. This allows for the swapping of specific parts within the framework without affecting other components or models. Modularity enables easy development of comparative systems designed to inform research questions above. The framework is structured to support a variety of experimental design approaches, including ablative tutor studies, tutor vs. traditional classroom training comparisons, intervention vs. non-intervention comparisons, and affect modeling and diagnosis research [6]. The descriptive use case illustrated next is based on an intervention comparison approach.

2 GIFT Experimental Use Case

In this section, we describe in detail the process of using GIFT to design and run a study to evaluate varying methods for communicating information to a learner while they interact with a game-based environment. This experiment was designed to examine varying modality approaches for feedback information delivery during a game-based learning event that is not implicit within the virtual environment (i.e., feedback in the scenario as a result of a player/entity or environmental change). This is influenced by available features present in the GIFT architecture and the benefits associated with research surrounding learning and social cognitive theory [10-11]. The notion is to identify optimal approaches for providing social agent functions to deliver feedback content that is cost effective and not technically intensive to implement. As a result, research questions were generated around the various communication modalities GIFT provides for relaying information back to the learner.

A functional component unique to GIFT is the Tutor-User Interface (TUI). The TUI is a browser-based user-interface designed for collecting inputs (e.g. survey and assessment responses) and for relaying relevant information back to the user (e.g. performance feedback). In terms of providing real-time guided instruction, the TUI can be used as a tool for delivering explicit feedback content (i.e., guidance delivered outside the context of a task that relays information linking scenario performance to training objectives) based on requests generated from the pedagogical model. Because the TUI operates in an internet browser window, it supports multimedia applications and the presence of virtual entities acting as defined tutors. As a potential driver for interfacing with a learner, research is required to evaluate feedback delivery in the TUI and assess its effectiveness in relation to other source modality variations. The overarching purpose of the described research is to determine how Non-Player Characters (NPCs) can be utilized as guidance functions while learning in a virtual world environment and to identify tradeoffs among the varying techniques.

Research questions were generated around the limitation associated with using the TUI during game-based instruction. For a virtual human to be present in the TUI, it requires a windowed display of the interfacing game so the browser can be viewed in addition to the game environment, which may take away from the level of immersion users feel during interaction; thus removing a major benefit with utilizing a game-based approach in education. Specifically, this study will assess whether explicit feedback delivered by NPCs embedded in a scenario environment has a significant effect on identified dependent variables (e.g., knowledge and skill performance, and subjective ratings of flow, workload, and agent perception) when compared to external NPC feedback sources present in the TUI. In terms of serious games, the current research is designed to address how the TUI can be utilized during game-based interactions and determine its effectiveness versus more labor intensive approaches to embedding explicit feedback directly in the game world.

This experiment was the first implemented use of GIFT functioning as a testbed for empirical evaluation. During the process of its development, many components had to be hand authored to accommodate the investigation of the associated research questions. This involved integration with multiple external platforms (e.g., serious game TC3Sim, the Student Information Models for Intelligent Learning Environments (SIMILE) program, and Media Semantics); development of scenarios, training objectives, assessments, and feedback; exploration of available avenues to communicate information; and representing these relationships in the GIFT schema. In the following subsections, we will review the process associated with each phase listed above.

2.1 Testbed Development

GIFT provides the ability to interface with existing learning platforms that don't have intelligent tutoring functions built within. In these games, learners are dependent on implicit information channels to gauge progress towards objectives. Integrating the game with GIFT offers new real-time assessment capabilities that can be used to provide learner guidance based on actions taken within the environment that map to associated performance objectives.

For the instance of this described use case, the serious game TC3Sim was selected as the learning environment to assess the effect of differing feedback modality approaches. TC3Sim is designed to teach and reinforce the tactics, techniques, and procedures required to successfully perform as an Army Combat Medic and Combat Lifesaver [7]. The game incorporates story-driven scenarios designed within a game-engine based simulation and uses short, goal-oriented exercises to provide a means to train a closely grouped set of related tasks as they fit within the context of a mission [8]. Tasks simulated within TC3Sim include assessing casualties, performing triage, providing initial treatments, and preparing a casualty for evacuation under conditions of conflict (ECS, 2012). For the purpose of the experiment, GIFT had to be embedded within TC3Sim for the function of monitoring performance to trigger feedback that would ultimately influence data associated with the dependent variables of interest.

This required pairing of the two systems so that GIFT could consume game state messages from TC3Sim for assessment on defined objectives, and for TC3Sim to consume and act upon pedagogical requests coming out of GIFT. For this to happen, a Gateway Module had to be authored that serves as a translation layer between the two disparate systems. The Gateway Module was also modified to handle feedback requests that were to be delivered by programs external to the game. This included integration with MediaSemantics, desktop and server software that provides character-based applications and facilitated the presence of a virtual human in the TUI that would act as the tutor entity. Following, enhancements to the Communication Module/TUI had to be employed to support the variations in feedback modalities.

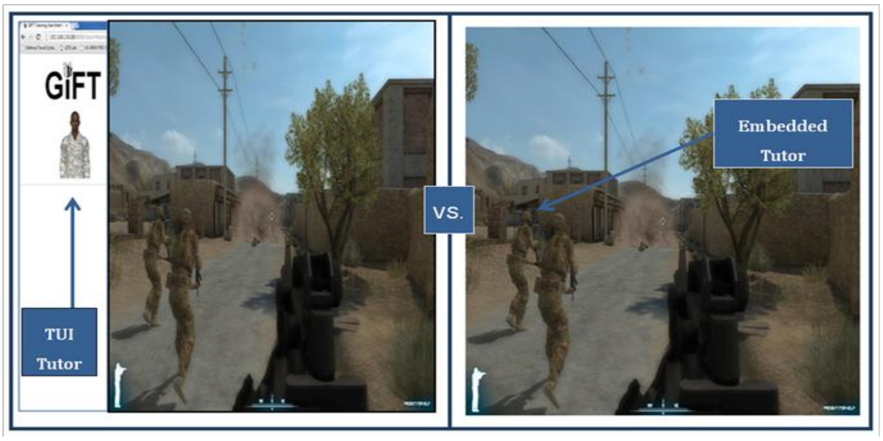


Fig. 4. Feedback Communication Modes

Communication Development. The functional component of GIFT primarily assessed in this research is the Communication Module/TUI, and focused on interfacing approaches for delivering feedback communications during a game-based learning event. For this purpose the major variations associated with the framework took place in GIFT's TUI, as well as identifying approaches for GIFT to manage agent actions within a virtual environment. This required two GIFT/TC3Sim versions with modifications to how the game was visually represented (see Figure 2). With a windowed version of the game, the MediaSemantics character was embedded into the TUI browser and was programmed to respond to feedback requests coming out of the domain module. Furthermore, two additional control conditions were authored to assess whether feedback delivered as audio alone made a difference and a condition with zero feedback to determine whether the guidance had any effect on performance. All participants interacted with the same scenarios, with two conditions including an EPA present in the virtual environment as an NPC. The remaining conditions will receive feedback from external sources to the game. With the functional modifications in place, the next step was designing scenarios, assessments, and feedback scripts.

Scenario Development. With the ability to apply varying techniques of feedback delivery during a game-based learning event, the next step was to design a scenario in TC3Sim to test the effects of all approaches. This requires multiple steps to ensure scenario elements are appropriate so that they lend to accurate inference based on the associated data captured during game interaction. This involved the definition of learning objectives the scenario would entail, associated assessments to gauge performance on objectives, and feedback to apply when performance was deemed poor.

Objectives were informed by competencies identified in ARL-STTC's Medical Training Evaluation Review System (MeTERS) program, which decomposed applied and technical skills for Combat Medics and Combat Lifesavers into their associated tasks, conditions, and standards for assessment purposes (Weible, n.d.). In development of the TC3Sim, the identified competencies were further decomposed into specific learning objectives in terms of enabling learning objectives and terminal learning objectives for each role and task simulated in the game environment. With guiding specifications, a scenario was developed that incorporated decision points for treating a hemorrhage in a combat environment. The scenario was designed to be difficult enough that participants would struggle, resulting in triggered feedback, while not being too difficult that successfully completing the task was impossible.

However, before explicit feedback linked to performance can be delivered in game-based environment, methods for accurately assessing game actions as they relate to objectives is required. The first step to achieve this is properly representing the domain's objectives within GIFT's Domain Knowledge File (DKF) schema by structuring them within the domain and learner model ontology. This creates a domain representation GIFT can make sense of, and results in a hierarchy of concepts that require assessments for determining competency. This association enables the system to track individual enabling objectives based on defined assessments, giving the diagnosis required to provide relevant explicit feedback based on specific actions taken. Following, methods for assessing the defined concepts must be applied that provide information for determining whether an objective has been satisfactorily met. For this purpose, ECS's Student Information Models for Intelligent Learning Environments (SIMILE) was integrated within GIFT.

Student Information Models for Intelligent Learning Environments (SIMILE). An innovative tool used in conjunction with TC3Sim for the purpose standardized assessment is SIMILE (ECS, 2012). In the context of this use case, SIMILE is a rule-engine based application used to monitor participant interaction in game environments and is used to trigger explicit feedback interventions as deemed by GIFT's learner and pedagogical models. In essence, SIMILE established rule-based assessment models built around TC3Sim game-state messages to generate real-time performance metric communication to GIFT. SIMILE monitors game message traffic (i.e., ActiveMQ messaging for this instance) and compares user interaction to pre-established domain expertise defined by procedural rules. As user data from gameplay is collected in SIMILE, specific message types pair with an associated rule authored and look for evidence determining if the rule has been satisfied; that information is then communicated to GIFT, which establishes if there was a transition in performance. Next, that performance state is passed to the learner model. GIFT interprets SIMILE performance metrics for the purpose of tracking progress as it relates to objectives. When errors in performance are detected, causal information is communicated by SIMILE in to GIFT, which then determines the feedback string to deliver.

Feedback Development. Following the completion of linking GIFT's domain representation with SIMILE-based assessments, specific feedback scripts had to be authored that would be presented when the pedagogical model made a 'feedback request'. In the design phase of these prompts, it was recognized that GIFT is dependent on a transition in performance before the pedagogical model can make any decision on what to do next. In the case of the TC3Sim scenario, this requires the player to perform certain actions that denote competency on a concept, but a question is, what information is available to determine they were ignoring steps linked to an objective?

From this perspective, it was recognized that time and entity locations are major performance variables in such dynamic operational environments. Outcomes in hostile environments are context specific, and time to act and location of entities are critical metrics that require monitoring. From there, if a participant had not performed an action in the game or violated a rule that maps to an associated concept, GIFT could provide reflective prompts to assist the individual on what action to perform next. An example applied in the experiment is 'Maintain Cover'. This requires staying out of the streets while walking through a hostile urban environment. For assessment, the player's distance from the street center was monitored, with a defined threshold designating if they maintained appropriate cover. For each concept, rules based on time and locations were identified, and reflective prompts were authored for each concept. Following, audio for each feedback prompt was recorded. This was the final step before the system could be fully developed.

3 Data Collection and Analysis Prep

Data collection was conducted over a five-day period at the United States Military Academy (USMA) at West Point, NY where a total of 131 subjects participated. This resulted in 22 participants for each experimental condition minus the control, which totaled at 21 subjects. The lab space was arranged for running six subjects at a time, with two experimental proctors administering informed consents and handling any technical issues that arose during each session. Once a subject logged in, GIFT managed all experimental procedures and sequencing, allowing the proctors to maintain an experimenter's log for all six machines. This feature shows the true benefit of GIFT in an experimental setting. Once properly configured, GIFT administers all surveys/tests and opens/closes all training applications linked to the procedure, thus reducing the workload on the experimental proctor and enabling multiple data sessions to be administered at a single time. GIFT offers the Course Authoring Tool (CAT) to create the transitions described above. A researcher can author the sequence of materials a participant will interact with, including transition screens presented in the TUI that assist a user in navigating through the materials.

Following the experimental sessions, data must be extracted from associated log files and prepped for analysis. A tool built into GIFT to assist with this process in the Event Reporting Tool (ERT). The ERT enables a researcher to pull out specific pieces of data that are of interest, along with options on how the data is represented (i.e., user can determine if they would like to observe data in relation to time within a learning event or to observe data between users for comparison). The result is a .CSV file containing the selected information, leaving minimal work to prepare for analysis. In this use case, the majority of analysis was conducted in IBM's SPSS statistical software, with the ERT playing a major role in the creation of the master file consumed by the program. This drastically reduced the time required to prep data for analysis, as it removed the need to input instrument responses for all subjects, it structured the data in a format necessary for SPSS consumption (i.e., each row of data represents an individual participant), and produced variable naming conventions listed on the top row.

4 The Way Ahead

GIFT provides a potent testbed in which studies of instructional techniques can be evaluated. Specifically, it allows researchers to investigate how best to implement tenants of intelligent tutoring, including optimal mechanisms for tracking performance, providing feedback and improving outcomes. At the current moment, GIFT provides limited feedback mechanisms that are generally used as formative prompts for correcting errors and reaffirming appropriate actions. New feedback approaches must be explored, such as natural language dialog, to expand the available options for relaying information in game environments. As well, research needs to identify approaches for using environmental cues in the game world to act as feedback functions informed by GIFT. In terms of GIFT as a testbed, advancements need to be applied to the ERT in terms of how data is exported to ease the required post-processing leading to analysis. This includes the ability to segment data in log files based around defined events in the environment that are of interest in analysis. Future research can build on the use case presented and/or conceptualize other investigations that benefit from GIFT.

5 References

1. Heylen, D., Nijholt, A., R., o. d. A., Vissers, M.: Socially Intelligent Tutor Agents. In: T. Rist, R. Aylett, D. Ballin & J. Rickel (Eds.), Proceedings of Intelligent Virtual Agents (IVA 2003), Lecture Notes in Computer Science, 2792: 341-347. Berlin: Springer (2008)
2. VanLehn, K.: The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. Educational Psychologist, 46(4): 197-221 (2011).
3. Porayska-Pomsta, K., Mavrikis, M., Pain, H.: Diagnosing and acting on student affect: the tutor's perspective. User Modeling and User-Adapted Interaction, 18(1): 125-173 (2008).
4. Mayes, T., Freitas, S. d.: Review of e-learning frameworks, models and theories. JISC e-Learning Models Desk Study, from http://www.jisc.ac.uk/epedagogy/ (2004).

5. Beal, C., Lee, H.: Creating a pedagogical model that uses student self reports of motivation and mood to adapt ITS instruction. In: Workshop on Motivation and Affect in Educational Software in conjuctions with the 12th International Conference on Artificial Intelligence in Education (2005).
6. Sottilare, R., Goldberg, B., Brawner, K., Holden, H.: Modular Framework to Support the Authoring and Assessment of Adaptive Computer-Based Tutoring Systems. In: Proceedings of the Inteservice/Industry Training, Simulation, and Education Conference, (2012).
7. S: vMedic. Retrieved from http://www.ecsorl.com/products/vmedic (2012).
8. Fowler, S., Smith, B., & Litteral, C.: *A TC3 game-based simulation for combat medic training* (2005).
9. Weible, J.: Preparing soldiers to respond knowledgeably to battlefield injuries. *MSTC U.S. Army*. (n.d.).
10. Bandura, A.: Social Cognitive Theory: An Agentic Perspective. Annual Review of Psychology, 52: 1-26. (2011).
11. Vygotsky, L.: Zone of Proximal Development. Mind in Society: The Development of Higher Psychological Processes: 52-91. (1987).

Authors

Benjamin Goldberg: is a member of the Learning in Intelligent Tutoring Environments (LITE) Lab at the U.S. Army Research Laboratory's (ARL) Simulation and Training Technology Center (STTC) in Orlando, FL. He has been conducting research in the Modeling and Simulation community for the past five years with a focus on adaptive learning and how to leverage Artificial Intelligence tools and methods for adaptive computer-based instruction. Currently, he is the LITE Lab's lead scientist on instructional strategy research within adaptive training environments. Mr. Goldberg is a Ph.D. Candidate at the University of Central Florida and holds an M.S. in Modeling & Simulation. Prior to employment with ARL, he held a Graduate Research Assistant position for two years in the Applied Cognition and Training in Immersive Virtual Environments (ACTIVE) Lab at the Institute for Simulation and Training. Mr. Goldberg's work has been published across several well-known conferences, with recent contributions to both the Human Factors and Ergonomics Society (HFES) and Intelligent Tutoring Systems (ITS) proceedings.

Jan Cannon-Bowers: is Director of Research at the Center for Advanced Medical Learning and Simulation (CAMLS) at the University of South Florida (USF) in Tampa, FL. She holds MA and Ph.D. degrees in Industrial/ Organizational Psychology from USF. She served as Assistant Director of Simulation-Based Surgical Education for the Department of Education at the American College of Surgeons from 2009 - 2011. Previously, Dr. Cannon-Bowers served as the U.S. Navy's Senior Scientist for Training Systems where she was involved in a number of large-scale R&D projects directed toward improving performance in complex environments. In this capacity she earned a reputation as an international leader in the area of simulation and game-based training, team training, training evaluation, human systems integration and applying the science of learning to real-world problems. Since joining academia, Dr. Cannon-Bowers has continued her work in technology-enabled learning and synthetic learning environments by applying principles from the science of learning to the design of instructional systems in education, healthcare, the military, and other high performance environments. She has been an active researcher, with over 150 publications in scholarly journals, books and technical reports, and numerous professional presentations.

Bringing Authoring Tools for Intelligent Tutoring Systems and Serious Games Closer Together: Integrating GIFT with the Unity Game Engine

Colin Ray, Stephen Gilbert

Iowa State University, Ames, IA, USA
{rcray, gilbert}@iastate.edu
http://www.iastate.edu

Abstract. In an effort to bring intelligent tutoring system (ITS) authoring tools closer to content authoring tools, the authors are working to integrate GIFT with the Unity game engine and editor. The paper begins by describing challenges faced by modern intelligent tutors and the motivation behind the integration effort, with special consideration given to how this work will better meet the needs of future serious games. The next three sections expand on these major hurdles more thoroughly, followed by proposed design enhancements that would allow GIFT to overcome these issues. Finally, an overview is given of the authors' cur- rent progress towards implementing the proposed design. The key contribution of this work is an abstraction of the interface between intelligent tutoring systems and serious games, thus enabling ITS authors to implement more complex training behaviors.

Keywords: intelligent tutoring, serious games, virtual environments, game engines

1 Introduction

Experience with the Generalized Intelligent Framework for Tutoring (GIFT) has shown that authoring new courses, domain knowledge, and learner configurations requires little-to-no programming experience. A basic understanding of XML and how the modules of GIFT interact is sufficient to design and configure a course for one of the supported training applications. When it comes to extending the framework to support new training applications, however, each interface module must be handcrafted. Reducing the amount of effort required to author a tutor and its content is a desirable quality of future authoring tools [1], therefore the task of integrating new training applications should be made as seamless as possible.

Serious games are one example of training applications that are well-suited for integration with ITSs; two such games are already supported by GIFT: Virtual Battlespace 2 (VBS2) and TC3 vMedic. These games encompass a only a subset of the training material that is possible with serious games, however. There are certain aspects of this genre of game are common across all individual applications, meaning that it may be possible to create a single abstraction layer capable of decoupling GIFT from the training application. This approach is recommended by Sottilare and Gilbert, who suggest that such an abstraction layer might be able to translate learning objectives into meaningful objects actions in the game world, and vice versa [2].

In addition to adapting data about the game state to a format that the ITS expects, it is also desirable for the ITS to have a finer degree of control over the scenario itself. These so-called “branching” or “conditional” scenarios [2] are difficult to achieve if the serious game and its plugin API are not designed with such functionality in mind. Therefore, it may also be necessary to “standardize” the ability to branch scenarios in the design of serious games.

To these ends, our proposed solution is to bring the ITS authoring tools closer to the content authoring tools used to create a given serious game. In the case of this paper, we have chosen to work with the popular Unity game engine. In the following sections we will show how integration with Unity and other serious game authoring tools can achieve the functionality that is currently desired in a modern ITS authoring suite.

2 Current Authoring Capabilities

As stated by Sottilare et. al, authoring new intelligent tutors is one of the three primary functions of GIFT [3]. To this end, the framework already contains authoring tools that enable users to create and configure the essential elements of an intelligent tutoring program. The following list gives a brief overview of the current authoring capabilities supported by GIFT:

- Authoring learner models through the Learner Configuration Authoring Tool (LCAT)
- Configuring sensors through the Sensor Configuration Authoring Tool (SCAT)
- Authoring Domain Knowledge Files (DKFs) through the DKF Authoring Tool (DAT)
- Creating and presenting surveys through the Survey Authoring Tool (SAT)

By using good design principles, the authors of GIFT have been able to effectively decouple the authoring of individual tutor components from one another. The decoupling of different program elements is important for improving the maintainability and extensibility of large pieces of software such as GIFT. One area of the framework design that suffers from tight coupling is the integration of third-party training applications, e.g. VBS2, vMedic, etc.

The development of these authoring tools is guided by several design goals, one of which is to “Employ standards to support rapid integration of external training/tutoring environments.” [3] In this regard, the current GIFT authoring construct can benefit from design enhancements that standardize this process across a range of training applications. Through the work outlined in this paper, we aim to generalize the process of integrating serious games with GIFT by creating an abstraction layer between GIFT and the game engine itself.

3 Related Work

Prior work in integrating serious games and intelligent tutors has demonstrated that ITS authoring tools can be easily adapted to work with individual games. Research conducted by Gilbert et al. demonstrated interoperation between the Extensible Problem-Specific Tutor (xPST) and a scenario created in the Torque game engine [4].

Devasani et al. built upon this work and demonstrated how an authoring tool for interpreting game state and player actions might be designed [5]. For their work, xPST was integrated with a VBS2 scenario. An important revelation made by the authors was that the author of the tutor need not define a complete state machine with transitions, since these transitions are implicit when the game engine changes state each frame.

Another of the GIFT design goals is to "Develop interfaces/gateways to widely used commercial and academic tools." [2] As previously mentioned, the current GIFT release has support for two serious games, one of which is VBS2, and the other being vMedic. This work and the previous two examples highlight the usefulness of integrating intelligent tutors with serious games, as well as the need for a standardized interface for authoring relationships between the game objects and tutor concepts. There are currently no concrete examples of a standard for quickly integrating serious games and intelligent tutors, although Sottilare and Gilbert make recommendations on how this problem might be approached [2].

4 Design Enhancements

As noted by previous authors [2, 4], one of the key challenges of tutoring in a virtual environment is mapping relevant game states to subgoals defined by the training curriculum. If the learner's goal is to move to a specific location, for example, the tutor author may not be interested in how the learner reached that state (e.g., driving, walking, or running). Thus, the tutor would have to know to filter out information from the game engine about modality of movement, and attend only to the location. If, however, the trainer wants to focus on exactly how best to move to that location (e.g., stealthily), then the tutor does need to monitor movement information. Using this example, we see that the context of the pedagogical goal influences the type of and granularity of tutor monitoring. From here on, we will refer to this challenge as the "observation granularity challenge."

In the process of reaching each pedagogical goal, the learner will build up a history of actions. Similar to the concept of a context attached to goals, there can also be context attached to patterns of actions over time. As an example, there may be cases where a tutor would permit errors in subgoals within a larger pattern of actions that it would still deem "successful." This history is essentially a recording of the virtual environment state over the course of the training. The amount and diversity of data in this history stream is potentially massive, creating a major challenge when attempting to recognize patterns. The problem of recognizing these patterns is crucial for identifying the learner's progress. From here on, we will refer to this challenge as the "history challenge."

In addition, because game environments afford interaction among multiple simultaneous entities, the tutor's reaction to actions and other new game states may be dependent on the actor. This context dependence suggests that it would be a valuable to add game entity attributes to state updates, and for GIFT to be able to process logic such as, "If the gunshot action came from an entity that is unknown or hostile, then take action X. If the gunshot came from a friend entity, take action Y." The additional layer of entity attributes adds complexity to authoring, but will be necessary for modeling team and social interactions. Devasani et al. describes a possible state-based architecture that could be the basis for such an approach, and it could be incorporated into GIFT [4]. From here on, we will refer to this challenge as the "actor-context challenge."

4.1 Abstraction Layer

A core aspect of the design principles behind GIFT is its generalizability to new training applications and scenarios. For this reason it is critical that the representations of data in GIFT and in the training application be allowed to remain independent. It is infeasible to force training applications to adapt to the interfaces that GIFT provides. However, a layer of abstraction that adapts messages from a sender into a form that can be consumed by a receiver is similar to the classic Adapter design pattern in software engineering. This design pattern has the useful property of enabling two otherwise incompatible interfaces to communicate, in addition to decreasing the coupling between them. In the case of GIFT, the abstraction layer would handle the mapping of objects from one service into a representation that makes sense to the other. As an example, this module might receive a message from the game engine containing the new location of the learner in the virtual environment which might then be interpreted for the tutor as "the learner reached the checkpoint."

In addition to mapping game engine concepts to tutor concepts, the abstraction layer can act as a filter in order to solve the observation granularity and history challenges. The scripting language achieves this by affording "do not care" conditions that would then trigger the abstraction layer to interpret only the relevant messages and discard everything else.

One proposed method for implementing this mapping is a scripting language and engine that allows the author to define the mapping themselves. Although it is far from being an automated solution, a scripting language would allow the ITS and content authors to hook into more complex behaviors with very little learning overhead. Scripting languages can be more user-friendly than XML by virtue of their syntactical similarities to written English. Furthermore, within the context of the Unity development environment we can expect users to have familiarity with scripting languages such as JavaScript and Boo (similar to Python). For these reasons, a scripting language is a natural choice for abstracting communication between GIFT and Unity. It is important for the simplicity of tutor authoring that this messaging abstraction layer have the tutor-side representation use language that a trainer would naturally use. If this is the case, the trainer can more easily author feedback and adaptive scenarios.

Although JavaScript and Boo are well-suited as tools to implement complex behaviors for game objects, they overcomplicate the task of describing interactions between the game world and the tutor. Instead of complex behaviors, we seek to enable the tutor author to quickly declare relationships between objects in the game, domain knowledge, and pedagogical goals.

In order to avoid burdening the author with the challenge of authoring different components in different languages, it may be advantageous to use XML for authoring abstraction layer rules. The declarative nature of XML makes it ideal for this role, although as mentioned previously, it suffers from poor readability. An alternative to XML is TutorScript, a scripting language developed for use in ITSs [6]. The design of TutorScript centers around the sequences of goals or contexts called a predicate tree. TutorScript's primary advantage over the previously mentioned alternatives is that it was designed with the goal of relating domain knowledge to learner actions in the training application. Additionally, TutorScript takes inspiration from Apple script in regards to syntax, allowing non-programmers to write scripts that read like English. For our work, TutorScript would allow us to hook into objects in both GIFT and Unity, where we can then create interactions using simple language.

4.2 Unity Editor

One of the main benefits of the Unity editor is that it is extensible to support user-created tools for custom workflows, or to fill in functionality lacked by the default editor. Some examples of editor plugins authored by users have added advanced level building tools, cut-scene editors, and even node-based visual scripting interfaces. The ultimate goal of this project is to completely integrate GIFT's authoring tools with the Unity ecosystem. This entails creating editor plugins for the entire suite of GIFT authoring tools, thereby enabling content authors to generate serious game and tutor content side-by-side using a single development environment.

An added benefit of integration with the Unity editor is its powerful rapid-prototyping abilities. Scenarios in Unity are organized into "Scenes" which can be loaded individually, played, and paused within Unity's built-in player. Current work to develop a proof-of-concept has demonstrated that it is possible to interact with the tutor within this player, thereby enabling the author to perform debugging on the training scenario to an extent.

It is considered good practice when authoring Unity games to "tag" game objects with names that encode the meaningful behavior that the game object performs. Assuming that the author adheres to this practice, the tagging mechanism combined with the abstraction layer will solve the actor-context challenge. Tags can be transmitted with game state updates that pass through the abstraction layer, which will then interpret the tags into context that is meaningful to the tutor. Since the abstraction layer is scripted by the author, it is essential for the abstraction layer script editor to be included in Unity's authoring suite. Making these tools easily accessible from one or the other allows the author to update changes to the scripts as soon as he or she makes changes to game object tags and other metadata.

As stated previously, the scripting languages provided by Unity may not be ideally suited to the task of communicating between the game engine and the tutor. Additional modifications will need to be made to MonoDevelop, the highly extensible IDE distributed as part of Unity, in order to support TutorScript or a variant of it. MonoDevelop greatly simplifies the creation of helpful programming tools such as syntax-highlighting and auto-completion that assist users with no prior programming background. Developing a MonoDevelop add-on for TutorScript also allows the author to more easily manage large or complex scripts needed to address the history and actor-context challenges via the built-in code organization features such as collapsing scopes. Taken together, Unity and MonoDevelop can be used as a suite of tools for authoring not only serious game content, but also advanced tutor behaviors, curriculum, and domain knowledge that will drive the training scenarios.

5 Recommendations

We project that the design enhancements recommended in this paper will assist in improving time savings and reducing cost involved with authoring an intelligent tutor. Specifically, we are aiming to reduce the time required to integrate GIFT with a new serious game by instead integrating it with the game engine itself. Our reasoning is that there are relatively few game engines that would need to be integrated, compared to the number of games with potential for enhancement through tutoring. Additionally, code reuse is facilitated by employing a standard format for describing relationships between game and tutor objects. If successful, this work will introduce a new abstraction layer between GIFT and the game engines that drive serious serious games, enabling a single tutor configuration to be deployed across a wide range of scenarios. For your convenience, the recommendations have been consolidated and figured in the table below.

Table 3. GIFT Design Enhancement Recommendations

Improve decoupling of potential learner actions and other game-specific data from the gateway and other GIFT modules.
Define a new XML schema for constructing game-tutor object relationships.
Develop a new authoring tool capable of authoring and validating these relationships.
Integrate new and existing authoring tools with the Unity editor.

6 Current Work

At this point we have successfully developed a proof-of-concept plugin that demonstrates basic communication between GIFT and Unity-driven games, similar to the interoperation module developed for VBS2. The extent of this functionality encompasses connecting to the Unity plugin from GIFT and then issuing playback commands such as pause and resume to the Unity player. This work has helped to increase our understanding of the inner workings of GIFT with regard to the augmentation required to communicate with our abstraction layer. In particular, the extent to which GIFT is tailored to each training application became apparent. In addition, we were able to leverage support for C# .NET 2.0 in Unity to move a great deal of the supporting code into components attached to game objects. This design allows the three services (Unity, Abstraction Layer, and GIFT) to remain isolated from one another during development, encouraging loose coupling across service boundaries and portability to other serious game authoring tools.

Before any work on the abstraction layer can begin, the language used to define object relationships must first be well-defined. Once this step is completed, we can begin abstracting away the elements of third-party application integration in GIFT that are currently hard-coded. Ultimately, these elements will be encapsulated by the proposed abstraction layer.

7 Conclusion

In this paper we proposed a handful of major design enhancements to GIFT with the overarching goal of bringing the ITS authoring workflow into the game content creation pipeline. The first task in realizing this vision is to create an abstraction layer comprised of a scripting engine tailored for ITSs. The second and final task is to integrate the GIFT authoring tools into Unity, in order to encourage side-by-side development of game and tutor content. The Unity game engine has been chosen for this work due to its ease of use, cross-platform support, and high extensibility. It is our hope that such a comprehensive suite of tools will help to drive a new generation of high-quality serious games.

8 References

1. Brawner, K., Holden, H., Goldberg, B., Sottilare, R.: Recommendations for Modern Tools to Author Tutoring Systems. (2012)
2. Sottilare, R.A., Gilbert, S.: Considerations for adaptive tutoring within serious games: authoring cognitive models and game interfaces. (2011)
3. Sottilare, R.A., Brawner, K.W., Goldberg, B.S., Holden, H.K.,: The Generalized Intelligent Framework for Tutoring (GIFT). Technical report (2013)
4. Gilbert, S., Devasani, S., Kodavali, S., Blessing, S.: Easy Authoring of Intelligent Tutoring Systems for Synthetic Environments. (2011)
5. Devasani, S., Gilbert, S. B., Shetty, S., Ramaswamy, N., Blessing, S.: Authoring Intelligent Tutoring Systems for 3D Game Environments. (2011)
6. Blessing, S.B., Gilbert, S., Ritter, S.: Developing an authoring system for cognitive models within commercial-quality ITSs. (2006) 497–502

Authors:

Colin Ray is a graduate student at Iowa State University, where he is pursuing an M.S. in Human Computer Interaction and Computer Science under the guidance of Stephen Gilbert, Ph.D. He possesses a B.S., also from Iowa State University, in the field of Electrical Engineering. His current research is focused on integrating intelligent tutoring systems with entertainment technology. In addition to ITS research, he is also conducting research and development in the areas of wireless video streaming and mobile surveillance to develop a platform for studying 360-degree video interfaces.

Stephen Gilbert, Ph. D., is the associate director of the virtual reality applications center (VRAC) and human computer interaction (HCI) graduate program at Iowa State University. He is also assistant professor of industrial and manufacturing systems engineering in the human factors division. His research focuses on intelligent tutoring systems. While he has built tutors for engineering education and more traditional classroom environments, his particular interest is their use in whole-body real-world tasks such as training for soldiers and first responders or for machine maintenance. He has supported research integrating four virtual and three live environments in a simultaneous capability demonstration for the Air Force Office of Scientific Research. He is currently leading an effort to develop a next-generation mixed-reality virtual and constructive training environment for the U.S. Army. This environment will allow 20-minute reconfiguration of walls, building textures, and displays in a fully tracked environment to produce radically different scenarios for warfighter training. Dr. Gilbert has over 15 years of experience working with emerging technologies for training and education.

Authoring a Thermodynamics Cycle Tutor Using GIFT

Mostafa Amin-Naseri[1], Enruo Guo[2], Stephen Gilbert[1], John Jackman[1], Mathew Hagge[3], Gloria Starns[3], LeAnn Faidly[4]

[1] Department of Industrial & Manufacturing Systems Engineering
[2] Department of Computer Science
3 Department of Mechanical Engineering
Iowa State University, Ames, IA, 50011 USA
4 Department of Mechanical Engineering
Wartburg College, Waverly, IA 50677 USA
{aminnas, enruoguo, gilbert, jkj, fforty, gkstarns}@iastate.edu,
leann.faidley@wartburg.edu

Abstract. The main idea of generalized intelligent tutoring system (ITS) development tools like Generalized Intelligent Framework for Tutoring (GIFT) is to provide authors with high-level standards and a readily reusable structure within different domains. Hence, adapting such a tool could be the best way to boost an underdeveloped tutor. In this paper we propose the design for a new GIFT-based tutor for undergraduate thermodynamics. An existing Thermodynamics Cycle Tutor has been designed that is meant to facilitate problem framing for undergraduate students. We describe the advantages of integrating this tutor with GIFT to add student models. Also an approach for evaluating the pedagogical performance of the GIFT-enhanced tutor is described.

Keywords: GIFT, intelligent tutoring system, thermodynamics cycle

1 Introduction

One of the most important challenges for engineering students is problem solving. Complex engineering problems typically contain multiple constraints, require multiple ideas, and may not have clear criteria for deciding the best solution. Beginning students struggle with engineering problem solving, and it has been observed that the initial stage (i.e., framing the problem) often causes the most difficulty. Students find it difficult to frame a complex problem, identify the core components, and brainstorm a possible solution path. These difficulties triggered the idea of building a tutor that can help undergraduate engineering students with their problem framing.

Thermodynamics cycles were our choices of topic to start with. In a National Science Foundation (NSF) funded project, a web-based software was developed to give students the ability to draw some initial sketches of the problem. Their drawing will be evaluated with regard to the expert model provided by the instructor and respectively they will be provided with different types and categories of feedback and instructions.

Regardless of how much effort is devoted to a project, there is always room for improvement. Key advantages of a generalized approach to ITS development (and GIFT in particular) are their standards and their high potential for reuse across educational and training domains. Other advantages that drive efficiency and affordability are GIFT's modular design and standard messaging; its largely domain-independent components; and its reuse of interfaces, methods, and tools for authoring, instruction, and analysis. Given these GIFT characteristics, there are many ways that the tutor could be enhanced being incorporating into GIFT. This will also provide us with an invaluable testbed to examine a GIFT-enhanced tutor with the existing one.

In the following sections, first a brief description of the tutor will be given and then an overview of the ways that the existing tutor can be enhanced by GIFT will be demonstrated. Finally a testing opportunity for the software will be described.

2 Current tutor

We would like to describe our current intelligent thermodynamics cycle tutor for engineering undergraduate courses. For the purpose of conceptualization and design, an ITS is often thought as consisting of several interdependent components: domain model, learner model, expert model, pedagogical module, interface and training media (Beck, Stern & Haugsjaa, 1996; Sottilare & Gilbert, 2011; Sottilare & Proctor, 2012).

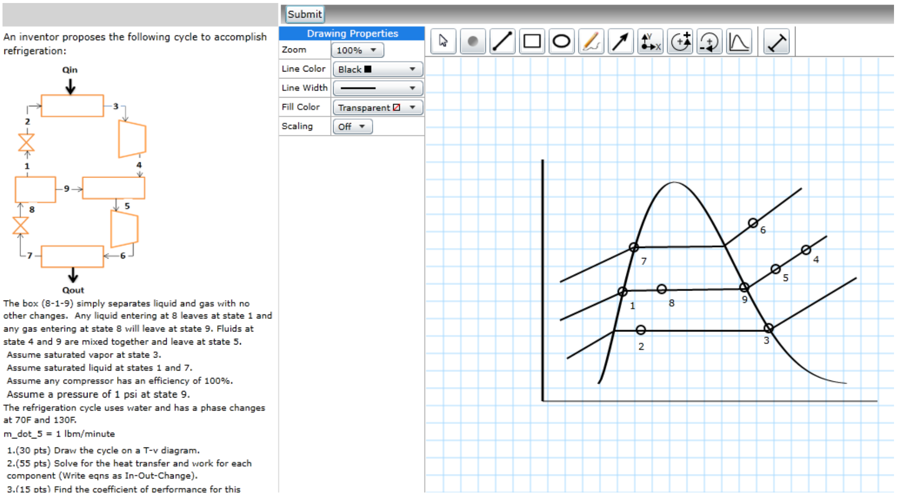
2.1 Domain model

The domain is about thermodynamics cycle problems. The goal is to understand how changes in pressure, temperature, specific volume and entropy interact with some commonly-used components, such as pump, compressor, turbine, expansion value, evaporator, heat exchanger, liquid-gas separator and mixing chamber. Based on the physical and chemical properties, a rule is associated with each component. For example, when an object goes through a pump, the pressure will increase, while the temperature and specific volume will increase slightly. In the final version, the author will have the option to modify the rules (e.g. to assume constant specific volume, or to test a student with a component that doesn't make physical sense). The table below shows the rules associated with other components. The domain model contains these rules.

Table 4. Rules for several components

Component	Pressure	Temperature	Specific Volume	Entropy
expansion valve	decease	decrease	Increase	Increase
evaporator	Same	same, increase	Increase	Increase
compressor	increase	Increase	decrease	same, increase
mixing chamber	same	between	between	between
condenser	same	decrease, same	decrease	decrease
Liquid -gas separator	same	same	between	between

2.2 Interface



An inventor proposes the following cycle to accomplish refrigeration:

The box (8-1-9) simply separates liquid and gas with no other changes. Any liquid entering at 8 leaves at state 1 and any gas entering at state 8 will leave at state 9. Fluids at state 4 and 9 are mixed together and leave at state 5.

Assume saturated vapor at state 3.

Assume saturated liquid at states 1 and 7.

Assume any compressor has an efficiency of 100%.

Assume a pressure of 1 psi at state 9.

The refrigeration cycle uses water and has a phase changes at 70F and 130F.

m_dot_5 = 1 lbm/minute

1.(30 pts) Draw the cycle on a T-v diagram.

2.(55 pts) Solve for the heat transfer and work for each component (Write eqns as In-Out-Change).

3.(15 pts) Find the coefficient of performance for this

Zoom	100%
Line Color	Black ■
Line Width	———
Fill Color	Transparent ☒
Scaling	Off

Fig. 5. A screenshot of Thermodynamics Cycle Tutor. The student reads the problem at left and solves it by constructing a vapor dome diagram at right.

Thermodynamics Cycle Tutor has been developed as part of a problem framing research project funded by the National Science Foundation. The tutor basically contains two parts. On the left side, it contains system/component diagram, problem description and questions. The right side uses a web-based drawing interface, XDraw, developed internally by author Jackman using the Microsoft Silverlight framework. XDraw supports basic drawing objects such as vapor dome, point, line, rectangle and vector as well as freehand drawing. It also provides facilities to allow students to label the states and insert text on the drawing. A backend database saves students' diagrams. XDraw communicates with tutor server via a TCP socket. Several message types are defined in order to differentiate what information would be checked and the next action should be taken.

When it starts, the left side shows the system diagram and problem description. Students can start problem framing by drawing a vapor dome (T-V diagram in this case) and use lines and points to represent pressure curve and state, respectively, and apply labels according to the system diagram. After clicking submit button, the diagram is sent to the tutor server, which checks a specific part based on the query message. The tutor then sends back the evaluation result and instruction for the next action as a returned message. Students may be directed to another interface based on their performance in the current stage. We will talk about the detailed sequences in the expert model.

2.3 Expert model

The expert model sets standards and compares learner actions to determine the progress. In the thermodynamics cycle domain, the expert model contains the following:

1. Check vapor dome present.
2. Check number of pressures.
3. Check number of states.
4. Check Pressure and Temperature relations in each of the components.

After the student submits the drawing, the tutor will check if the drawing contains the vapor dome. If so, it will continue to the next check: number of pressures. If it is wrong, the students will be asked questions like, "How many pressures are there in the system?" showing on the left panel. If the student's answer is wrong, the tutor will go through all the components, and ask the pressure change within each of them. Some tutorial videos and illustrations will be provided to help them better understand the concept.

The content on the left panel will be changed based on the student's activity in a particular problem. For example, in the drawing, the student draws state 4 to the right of state 3. A compressor pushes the gas molecules closer together, so specific volume should decrease. The left panel will show a compressor's diagram, along with some questions, such as "How does the specific volume change in a compressor?" It contains several choices that students can select. If the student chooses the correct answer, it will ask the student to correct it in the diagram. If the student gets the wrong answer, it will direct the student to some tutorial video files and ask again.

2.4 Training media

In order to help students correct their misconceptions, the tutor provides some video files that include class lectures and illustration videos at a certain stage of the activity. The video files will be loaded automatically to ask students to watch when their answer is wrong. Generally speaking, the training media is domain-dependent and requires the instructor to prepare and pre-define what stage it should appear.

2.5 Learner model

Currently there is no learner model in our Thermodynamics Cycle Tutor. We think it is a good idea to monitor and keep track of students' current progress, save students' previous performances, and perform surveys. An example could be when student starts a new problem, the tutor should be able to select an appropriate problem from the learner model and predict how successful the student will be based on his/her historical data. Also, in the survey part, the tutor could receive feedback on the learner's background knowledge and quality of the pedagogical process. We believe GIFT could allow us to build a learner model easily, and we would like to explore how it may benefit our tutor.

2.6 Pedagogy

As a pedagogical learning tool, the tutor also needs to set up learning goals and pace for the students, so the student can learn each component's P, T, and V behavior one at a time (starting with the easiest one, and increasing difficulty as easier ones are mastered). The ideal tutor would be able to connect with other thermodynamic understanding, using ideas such as rate of heat transfer and rate of work (power) to connect with P, T and V relationships. For students with different performance levels, the problem difficulty should vary. The tutor's feedback has to inspire their thinking, not give them answers directly. The pedagogy module requires much flexibility and should vary based on different problem sets and instructor-student needs.

3 GIFT-enhanced tutor

The existing tutor is expected to demonstrate an acceptable functionality; however, there are limitations in its domain independence and reusability, and it also lacks some desirable features such as a learner model. Mitigating these limitations will require a considerable amount of time and programming effort. GIFT offers many features that can attenuate the level of programming skill and time required. Also, providing standards and well-defined domain independent structures facilitates the tool enhancement. The main benefits of GIFT for our tutor are explained below.

3.1 Learner model

A highly desired feature for intelligent tutors is to provide learners with personalized education (Woolf, 2010). In other words, if we could know the exact skills that learners do and do not have, then we could provide them with the exact resources they need. Learner model is a module that has been developed for this reason. Learner model keeps record of many aspects of the learner, such as the learner's progress toward objectives, actions taken in the interface and historical data (e.g., previous performance) (Sottilare, 2012). There is also a need to define some skill levels with respect to the learner's patterns. Having this valuable information about the learner and their skill, the tutor will be able to provide him or her with specific problems, feedback, instructional content, etc.

In our current tutor there are many data streams that are monitored (e.g., the mistakes or feedback types, instructional content provided, etc.). Also, by handing out surveys, some information about knowledge background is available. The problem is they are stored in separate databases and it is hard to put them together. Putting these data together can help us build the student model. GIFT provides the ability to store this data in a well-structured way, as it has the option for sensor data storing. In addition, we can benefit the GIFT survey authoring tool, to conduct our surveys in the same program and store them easily in the proper place. In this way, by defining the skill levels we will be able to build our learner models based on the data we have collected from them.

Another important feature is data reporting. Having collected a considerable amount of data on the learner, an easy-to-access way to extract knowledge out of it is necessary. The GIFT event report tool provides a proper interface to easily let users (instructors) access the data they desire.

For any further research, we might want to use different types of sensors to evaluate a learner's cognitive load or status or stress. GIFT provides the ability to readily acquire that data as well.

3.2 Multiple scenarios

Once skill levels have been assigned to learners, appropriate content must be provided based on those skills. To handle various types of problems and instructional material (i.e., Domain Knowledge Files), a precise structure is needed to store them. For this, GIFT Domain Authoring Tool (DAT) will be used. Since this tool can be used without having to launch the entire GIFT architecture, it enables us to benefit from it earlier in the development process.

In addition, different instructors have different pedagogical strategies and instructional content. Thus, they may want students to go through a different scenario or visit different content. Two of our co-authors, for example, have different pedagogical preferences for teaching the thermodynamic cycle. Based on their preferences, GIFT could enable us to create multiple scenarios appropriately. Without having a perfect match between the knowledge database and the tutor, accommodating multiple approaches would not be possible. However, GIFT has already provided the structured database, so making the linkage between the tutor and GIFT DAT will be helpful.

3.3 Expansion to other domains

The domain-independent structure of GIFT will enable us to simply customize our tool for different fields. Currently, statics problems, e.g., free body diagrams, can also be tutored via our tool, but using the Domain Authoring Tool that will facilitate the deployment of instructional material. The entire process of student model and learner-specific instructions could be implemented with this approach as well.

4 Proposed evaluation experiment

In the Fall 2013 semester, a thermodynamics class will be offered for undergraduate mechanical engineering students in Iowa State University. Early in the semester they will be divided into three groups. One group will work with the GIFT-enhanced tutor, another group with the existing tutor (without student model), and the last group will just join the class and have no tutor. Keeping records of the three groups' performances during the semester with periodic quizzes, as well as gathering data on their skills and solution time, will provide us with a valuable data to evaluate the performance of an intelligent tutor with the student model (GIFT-enhanced). It will also help us examine the effectiveness of the existing tutor.

5 Conclusion

After analyzing the features of our existing tutor and GIFT, they seem to complement each other perfectly and provide a comprehensive ITS. Using GIFT's standards for structuring the tutor, as well as data and file storing, will attenuate the requisite programming skill and effort to accomplish the same objectives. Also, its high domain-independence will create opportunities to expand the tutor to different learning domains. The GIFT-enhanced tutor will be compared with the existing tutor and with traditional class training during the 2013 Fall semester. The results could provide a documented comparison between two different methods of ITS development.

6 References

1. Beck, J., Stern, M., and Haugsjaa, E. (1996). Applications of AI in Education, ACM Crossroads.
2. Woolf, B. P. (2010). A Roadmap for Education Technology. National Science Foundation.
3. Sottilare, R. (2012), Adaptive Tutoring & the Generalized Intelligent Framework for Tutoring (GIFT), Retrieved from: http://www.ist.ucf.edu/grad/forms/2012_10_Sottilare_UCF_GIFT%20brief.pdf.
4. Sottilare, R. and Gilbert, S. (2011). Considerations for tutoring, cognitive modeling, authoring and interaction design in serious games. Authoring Simulation and Game-based Intelligent Tutoring workshop at the Artificial Intelligence in Education Conference (AIED) 2011, Christchurch, New Zealand, June 2011.
5. Sottilare, R. and Proctor, M. (2012). Passively classifying student mood and performance within intelligent tutoring systems (ITS). Educational Technology Journal & Society. Volume 15, Issue 2.

Authors

Mostafa Amin-Naseri: is a Master of Science student in Industrial and Manufacturing Systems Engineering in Iowa State University. His BS was in industrial engineering with a major in systems engineering. Having had an experience in tutoring high school and undergraduate students he got familiar with common mistakes and issues that students usually face when solving problems and also the necessity for personalized instructional material. This background led him to start working on Intelligent Tutoring Systems (ITS). He is currently working with a team on an ITS that helps undergraduate engineering students with problem framing in Statics and Thermodynamics. Applying statistical analysis and data mining techniques to learners' historical data in order to come up with learner models, evaluate skill levels and to offer customized instructional material and feedback, is one of his fields of interest. Finally, with a systems engineering background, he is also interested in analyzing and simulating the learning process using System Dynamics models.

Enruo Guo: a Ph.D. student in Computer Science co-majoring in Human Computer Interaction in Iowa State University. She got her Master's degree in Computer Science from Iowa State in 2012. She also had a background in pedagogy and psychology in her undergraduate study in Beijing Normal University, which is best-known for training school teachers in China. Her philosophy is to use computers to simplify human's learning process and make everything as simple as possible. She has broad interests in intelligent tutoring system, artificial intelligence, computer vision and virtual reality. She has strong enthusiasm in developing real-world applications to assist undergraduate teaching and administration. She develops Thermodynamics Cycle Tutor and now is working on Free-Body Diagram Tutor for engineering undergraduates. Furthermore, in order to reduce the workload of human inspector of Department of Chemistry, she develops Intelligent Safety Goggle Detector which can automatically detect if a lab user wears safety goggle at the entrance of the lab.

Stephen Gilbert, Ph.D., is the associate director of the virtual reality applications center (VRAC) and human computer interaction (HCI) graduate program at Iowa State University. He is also assistant professor of industrial and manufacturing systems engineering in the human factors division. His research focuses on intelligent tutoring systems. While he has built tutors for engineering education and more traditional classroom environments, his particular interest is their use in whole-body real-world tasks such as training for soldiers and first responders or for machine maintenance. He has supported research integrating four virtual and three live environments in a simultaneous capability demonstration for the Air Force Office of Scientific Research. He is currently leading an effort to develop a next-generation mixed-reality virtual and constructive training environment for the U.S. Army. This environment will allow 20-minute reconfiguration of walls, building textures, and displays in a fully tracked environment to produce radically different scenarios for warfighter training. Dr. Gilbert has over 15 years experience working with emerging technologies for training and education

Dr. John Jackman: an Associate Professor, Industrial and Manufacturing Systems Engineering at Iowa State University, conducts research in manufacturing and engineering education. In manufacturing, he is currently working on wind turbine blade inspection techniques that characterize the variability in blade geometry and detect surface flaws. Dr. Jackman has extensive experience in computer simulation, web-based immersive learning environments, and data acquisition and control. His work in engineering problem solving has appeared in the Journal of Engineering Education and the International Journal of Engineering Education. He is currently investigating how to improve students' problem framing skills using formative assessment.

Dr.Mathew Hagge: has built a teaching style for thermodynamics that simplifies the course into a small set of ideas and skills, and asks students to develop and apply these same ideas to novel situations. The same set of ideas and skills are used for every problem. No equation sheets or similar problems are used. Memorization is not needed, and will actually decrease student performance. Students are asked to make as many decisions as possible, subject to their level of understanding. As student knowledge and expertise increases, so does the problem complexity. Less than a dozen problems will be needed, but each new problem will push the student's understanding. By the end of the course, successful students have the skills to solve any problem in a traditional textbook, and to correctly solve problems much more complex than a traditional textbook. When students need help, Dr. Hagge has developed a set of questions that can identify the specific misunderstanding, and then provide an activity or discussion that will eliminate the misunderstanding.
Dr Hagge's teaching method is ideally suited for implementation with a tutor that focuses on student understanding. The tutor can measure specific skills/understanding and provide feedback unique to that student.

Dr Gloria Starns: received her Ph.D. in Mechanical Engineering from Iowa State University in 1996 and began instructing engineering students as a graduate student at Iowa State in 1990. Dr. Starns' interest in working with a personal tutoring system is related to her past work with concept based learning, as well as understanding the role that use of active and constructive learning has in enabling students to retain and use acquired knowledge; her role in this project has been to provide the research team problems of varying complexity for purposes of collecting data from the tutor as it continues to evolve.

Dr.LeAnn Faidley: is an Assistant Professor of Engineering Science at Wartburg College in Waverly, IA. She has a BS in Engineering Science and Physics from Iowa State University, an MS in Engineering Mechanics, and a MS and PhD in Mechanical Engineering from The Ohio State University. At Wartburg, Dr. Faidley teaches the freshman labs, the Engineering Mechanics sequence, the Design sequence, and Engineering Materials. She is interested in improving student engagement with engineering subjects through active learning, relevant projects, and interactive online tools.

Integrating GIFT and AutoTutor with Sharable Knowledge Objects (SKO)

Benjamin D. Nye

Institute for Intelligent Systems
University of Memphis, Memphis, TN 38111
benjamin.nye@gmail.com

Abstract. AutoTutor and the Generalized Intelligent Framework for Tutoring (GIFT) are two separate projects that have independently recognized the need for greater interoperability and modularity between intelligent tutoring systems. To this end, both projects are moving toward service-oriented delivery of tutoring. A project is currently underway to integrate AutoTutor and GIFT. This paper describes the Sharable Knowledge Object (SKO) framework, a service-oriented, publish and subscribe architecture for natural language tutoring. First, the rationale for breaking an established tutoring system into separate services is discussed. Secondly, a short history of AutoTutor's software design is reviewed. Next, the design principles of the new SKO framework for tutoring are described. Finally, the plans and progress for integration with the GIFT architecture are presented.

Keywords: Intelligent Tutoring Systems, Service Oriented Architectures, Message Passing, Design Patterns, Systems Integration

1 Introduction

Intelligent tutoring systems (ITS), despite effectiveness as instructional technology, have historically suffered from monolithic design patterns (Murray, 2003). Roschelle and Kaput (1996) referred to tutoring systems as "application islands" for their lack of interoperability. A recent systematic literature review by the author of this paper found little evidence of newer tutoring systems sharing components or working toward a common base of components (Nye, 2013). This lack of modular ITS services reduces the availability of ITS software by preventing sharing of ITS components between systems. This problem increases the cost of ITS development and imposes a high barrier to entry for new systems.

An improvement over this design would be a component-based and service-oriented architecture, allowing composability of ITS components. Composability would greatly benefit ITS research, due to the high interdisciplinary skill-set needed to build a full tutoring system. Service oriented design would allow specialists to focus on individual components, while sharing common components. It would also greatly reduce the waste of reimplementing components that could be shared by ITS. However, this concept is not new. Roschelle and Kaput (1996) suggested component-based design over a decade ago, but little meaningful progress has been made toward that end. Part of the problem was the relative novelty of tutoring systems: fewer established examples existed and there was less consensus about the definition and functionality of an ITS.

More recently, central researchers have noted that different ITS tools share many of the same high-level behaviors (VanLehn, 2006; Woolf, 2009). This consensus implies a common ontology for describing the high level functions of ITS components and the meaning of information passed between them. While literature consensus does not constitute a formal ontology, it indicates the possibility of a grammar for talking about the types of information communicated between different parts of an ITS. An argument against the feasibility of this approach might be the disconnected nature of many subfields of ITS research, which come from different theoretical backgrounds that are not easily integrated (Pavlik and Toth, 2010). With that said, regardless of the underlying theory, the external behaviors (e.g., giving a hint) and core assessments (e.g., learning gains) are quite similar. The need to maintain theoretical coherence does not mean that a common ontology is infeasible, but simply indicates that there are limits to its useful granularity. For example, does a user-interface care how a hint is generated? If not, the user interface should be able to display hints from any system capable of generating hints. By taking advantage of the distinct roles and functions within a tutoring system, breaking down an individual tutoring system into distinct, sharable components is possible. Moreover, a significant number of components of the tutoring system are secondary to the tutor's theoretical concerns but pivotal to their operation. Machine learning algorithms, data storage interfaces, facial recognition software, speech synthesis, linguistic analysis, graphical interfaces, and tutoring API hooks for 3D worlds are enabling technologies for tutoring systems (Pavlik et al., 2012; Nye et al., 2013).

AutoTutor and the Generalized Intelligent Framework for Tutoring (GIFT) are two separate tutoring frameworks that have independently recognized the importance of modularity and interoperability in tutoring design. AutoTutor is a highly-effective natural language tutoring system where learners talk through domain concepts with an animated agent (Graesser et al., 2004a). Learning gains for AutoTutor average 0.8σ over reading static text materials on the same topic (Graesser et al., 2012). GIFT is a service-oriented framework for integrating tutoring capabilities into static material, such as a PowerPoint, and interactive environments, such as a simulation or a serious game (Sottilare et al., 2012). This paper describes the process of moving AutoTutor toward a service-oriented paradigm and the progress toward integrating AutoTutor with GIFT.

2 Prior AutoTutor Design Patterns

The original AutoTutor design was implemented as a standalone desktop application to teach computer literacy, which also relied on platform-dependent elements such as the Microsoft Agent (Peter Wiemer-Hastings et al., 1998). Since an installed application made AutoTutor harder to deliver, a subsequent version reimplemented the tutoring system as a web-based application (Graesser et al., 2004a). Since that time, various tutoring systems that followed in AutoTutor's footsteps have used a mixture of desktop and web-based designs. While many of these systems share conceptual principles and some share authoring tools, reuse of components and services between these different tutoring projects has been limited. So then, while Roschelle and Kaput (1996) spoke of "application islands," AutoTutor and related systems have evolved as a sort of "application archipelago" of related but independent tutoring systems. While the principles of AutoTutor have been influential, code reuse has been limited, even in projects that explicitly extend AutoTutor, such as AutoTutor Lite (Hu et al., 2009).

AutoTutor's package that handles linguistic analysis is a counter-example to this pattern. Coh-Metrix provides a suite of linguistic analysis tools, such as latent semantic analysis, regular expression matching, and domain corpora (Graesser et al., 2004b). While this tool started development nearly a decade ago, it remains under active development and is used regularly by AutoTutor and other projects. This longevity may be attributed to its focused scope and purpose as a toolbox for linguistic analysis. Additionally, Coh-Metrix has the advantage that it is primarily algorithmic and algorithms do not tend to change much.

By comparison, the landscape of educational computing has changed greatly over that period: web-based applications replaced many desktop applications, then full-featured Java web applications were replaced by lighter JavaScript and Flash clients with server-side code written in languages such as Python and C#. AutoTutor designs have mirrored these trends fairly closely, with the original AutoTutor written as a desktop application (Peter Wiemer-Hastings et al., 1998), the next iteration being a Java-based web application (Graesser et al., 2004a), and systems such as AutoTutor Lite relying on Flash, JavaScript, and Python (Hu et al., 2009). In the process of changing platforms and programming languages, a great deal of development work has been lost to a cycle of re-implementation to match the needs of a changing technology landscape.

Based on this history, how could design patterns be improved to encourage reuse and interoperability? The first principle, demonstrated by Coh-Metrix, is embodied by the Unix philosophy: "Do one thing and do it well" (Raymond, 2003). This is fundamental to service-oriented design, where boundaries between components are strict. The second principle is that delivery platforms may evolve rapidly. Just as AutoTutor has adapted to web delivery for desktops, mobile applications are becoming an important platform. Tutoring systems need to minimize platform-dependence. Finally, the best programming languages for different platforms vary. Moreover, existing tutoring systems have large investments in their code base. Components need to communicate using language-agnostic standards for different tutoring systems to interoperate. Service-oriented designs, while not yet common in tutoring systems, offer significant advantages for the next generation of ITS.

3 Sharable Knowledge Objects

AutoTutor is moving in this direction with Sharable Knowledge Objects (SKO), which allow creating tutoring modules by composing a mixture of components: local static media, remote static media, local components, and web services. These components are categorized in terms of two questions: 1. Is the component local? and 2. Is the component static or interactive? While the current focus of this work is on service-oriented web delivery, the design is also intended to support communication between components in the same process. By using a uniform messaging pattern, components can be developed without consideration of whether they will be used on a local device or accessed as a remote service.

In design pattern terms, SKO's are being developed to follow the service composition principle. In service composition, a composition of multiple services can be considered a single service when creating a new composition of services. Service-oriented design is largely the same concept as component-based design, except with the added complexity that the components may be distributed across time and space as part of a distributed network. So then, what is a SKO? A SKO declares a composition of services intended to deliver knowledge to a user, with the expected use case being tutoring in natural language. In this context, the SKO framework is not a re-implementation of AutoTutor but a framework for breaking AutoTutor down into minimal components that can be composed to create tutoring modules that may or may not rely on the traditional AutoTutor modules. These minimal components are intended to be used as part of a service-oriented design.

Figure 1 shows an overview of the new SKO framework. The core of the new SKO framework relies on a publish-and-subscribe architecture based entirely on passing messages that convey semantically-tagged information. These patterns significantly improve the flexibility of service composition for tutoring. Publish and subscribe frees individual components from explicit knowledge of any other services. The component knows only its own state, the messages that it has received, and the messages that it has transmitted. SKO is viewed as a way to split AutoTutor into separate, easily-reusable components. Secondly, SKO is intended to unify components from different systems that have evolved from AutoTutor along divergent paths by adding their unique functionality as services.

Exploring the details of each of these services is outside the scope of this paper. Instead, this section will focus on how different users would interact with and benefit from a SKO. While certain features of SKO are still under development, these examples describe how different users will interact with the completed SKO framework. To the learner, a SKO acts as a single module of instructional content focusing on a single lesson (e.g., learning how to complete a given math problem). For AutoTutor Lite, a web page loads a talking head and a user-input box, often with a button to begin a tutoring session. The SKO module does not specify any rules or functions. Instead, it relies on components to send messages. So then, user input triggers on the tutoring button generates a message from the user interface component. The tutoring engine reads that message and selects tutoring dialog, which is sent off as a new message. The animated agent and text-to-speech services read this message and cause the talking head to speak the message to the learner. By sharing a student model in a learning management system, multiple SKO can be combined into larger lesson units.

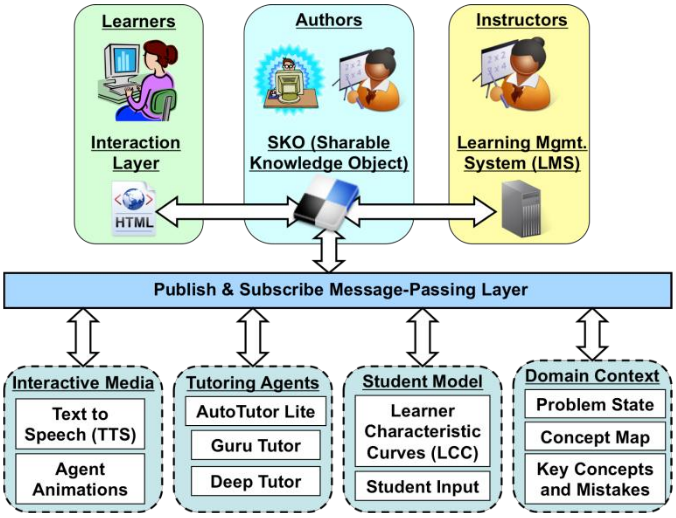


Fig. 6. Sharable Knowledge Object Framework for AutoTutor

To an advanced developer, a SKO is a collection of services. Advanced developers design new services and create SKO templates that can be filled in by instructors. These designers can create a SKO template using an advanced interface, where they would define the set of services within a SKO template and how these tie into the user interface. However, the advanced developer is not expected to add any domain content. Instead, they merely specify placeholders for content that is required or allowed. Based on these placeholders, a form-based authoring wizard would be created to allow instructors and domain experts to create specific SKO based on the template.

To an instructor, a SKO is a series of forms where they enter their expert data and produce working tutoring modules that they can test immediately. For example, an advanced developer could make a SKO template for guiding a student through solving an Algebra problem. From this, a form would be generated to allow an instructor to specify solution steps and tutoring dialogs associated with each step. An instructor could complete this form multiple times to enter content for different problems. This development is intended to be collaborative. By storing SKO in cloud hosts, different authors can edit or test each module. This also greatly facilitates SKO delivery, as a web-based SKO can be directly tested after creation.

4 Integration with GIFT

As part of the project to integrate AutoTutor with GIFT, AutoTutor Lite is being broken down into distinct services to fit into the SKO framework. Rather than focus on the low-level details of how AutoTutor and GIFT are integrating, the high-level process will be outlined. There is no canonical set of services that a given tutoring system should be broken down into so that it can be integrated into GIFT. However, the general integration process followed by AutoTutor might serve as a model for other systems considering GIFT integration. This integration has five phases: 1. identifying complementary functionality, 2. determining distinct “parts” of the AutoTutor Lite system, 3. specifying the functionality, inputs, and outputs of each part, 4. building web services, and 5. working with GIFT developers to add these to the GIFT distribution.

In the first phase, to identify complementary functionality between GIFT and AutoTutor, a large table of various key features for each system was created. This table helped identify the tools that GIFT had already implemented and those that AutoTutor Lite could contribute. This process identified that AutoTutor’s main contributions were conversational pedagogical agents, interactive tutoring, improved student modeling, and semantic analysis tools to compare sentence similarity. In the second phase, the full AutoTutor Lite system was examined to find distinct parts: sets of functionality that could be meaningfully split into distinct components. GIFT is meant to be a generalized system, so re-usable components offer more value to the system. To find these divisions, we looked for parts that only needed and returned small, well-formed information from other parts (e.g., the semantic service can compare any two sets of words and return a similarity value). In the next phase, the functions, inputs, and outputs of each part were determined. After that, we started building web services for each part. Web services were used because they follow communication standards that mean that AutoTutor code does not need to be in the same language as the GIFT code, nor does it need to run on the same computer. Finally, as versions of these web services have been completed, they have been provided to GIFT for integration into the system. This is an important part of the process, as testing with GIFT has helped uncover hurdles about the scalability and limitations of these new services. As these services are completed, they are being integrated into releases of GIFT.

Overall, integration with GIFT dovetails with a larger movement of AutoTutor toward a service-oriented architecture. This redesign will not only help integration with GIFT, but also with other systems in the future. Figure 2 shows how AutoTutor services are expected to integrate into the GIFT framework. AutoTutor services are shown on the right side of the diagram and include the semantic analysis service (for analyzing user input), learner’s characteristic curve (LCC) service (a simple type of student model), tutoring service for AutoTutor Lite, a service for text-to-speech, and an animated agent service. Some of these components are already available as web services. Once these services are available, GIFT will be able to incorporate basic AutoTutor Lite tutoring as part of its framework. The message-passing SKO framework will then standardize how AutoTutor communicates with GIFT. Additional services not displayed are also anticipated, such as a persistent student model, authentication service, and services for wrapping assessments such as multiple choice tests.

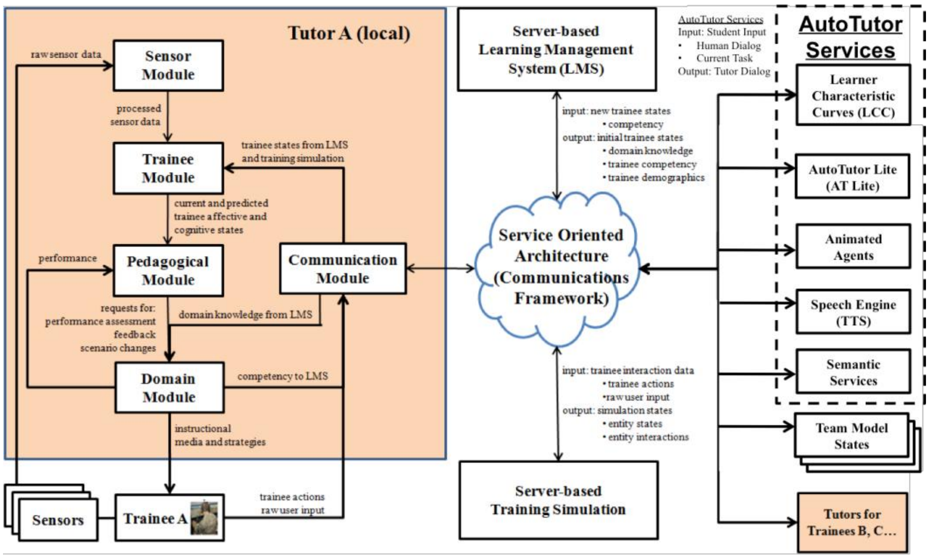

Tutor A (local)
raw sensor data
Sensor Module
processed sensor data
Trainee Module
trainee states from LMS and training simulation
current and predicted trainee affective and cognitive states
performance
Pedagogical Module
Communication Module
requests for: performance assessment feedback scenario changes
domain knowledge from LMS
Domain Module
competency to LMS
instructional media and strategies
Sensors
Trainee A
trainee actions raw user input
Server-based Learning Management System (LMS)
input: new trainee states
• competency
output: initial trainee states
• domain knowledge
• trainee competency
• trainee demographics
Service Oriented Architecture (Communications Framework)
input: trainee interaction data
• trainee actions
•raw user input
output: simulation states
• entity states
• entity interactions
Server-based Training Simulation
AutoTutor Services
Input: Student Input
• Human Dialog
• Current Task
Output: Tutor Dialog
AutoTutor Services
Learner Characteristic Curves (LCC)
AutoTutor Lite (AT Lite)
Animated Agents
Speech Engine (TTS)
Semantic Services
Team Model States
Tutors for Trainees B, C…

Fig. 7. Integration of AutoTutor and GIFT

5 Limitations and Future Directions

The SKO framework is intended to separate components based on the knowledge transferred between them, represented as semantic messages. This process will greatly improve modularity, enable AutoTutor to be implemented using a service-oriented design, and support interoperability with GIFT. However, modularity is limited by the information each component must share. Certain functions of the tutoring system are more easily separated into distinct components than others. For interoperating with additional tutoring systems, agreeing on a common set of messages may also be a challenge.

Currently, the publish-and-subscribe version Sharable Knowledge Object framework is under active development. In parallel with this work, AutoTutor Lite is being broken down into services and consumed by GIFT using traditional API's. Work in this area is focused on converting the semantic analysis services and AutoTutor Lite tutoring interpreter into services. Message-passing interfaces will then be incorporated into each service and they will be composed using the publish-and-subscribe SKO framework.

Acknowledgements The core SKO architecture is sponsored by the Office of Naval Research STEM Grand Challenge. Integration of AutoTutor Lite services with GIFT is sponsored by the Army Research Lab.

6 References

1. Graesser, A.C., Conley, M.W., Olney, A.: Intelligent tutoring systems. In: Harris, K.R., Graham, S., Urdan, T., Bus, A.G., Major, S., Swanson, H.L. (eds.) APA Educational psychology handbook, Vol 3: Application to learning and teaching, pp. 451–473. APA, Washington, DC (2012)
2. Graesser, A.C., Lu, S., Jackson, G.T., Mitchell, H.H., Ventura, M., Olney, A., Louwerse, M.M.: AutoTutor: A tutor with dialogue in natural language. Behavior Research Methods, Instruments, and Computers 36(2), 180–192 (2004a)
3. Graesser, A.C., McNamara, D.S., Louwerse, M.M., Cai, Z.: Coh-Metrix: Analysis of text on cohesion and language. Behavior Research Methods, Instruments, and Computers 36(2), 193–202 (May 2004b)
4. Hu, X., Cai, Z., Han, L., Craig, S.D., Wang, T., Graesser, A.C.: AutoTutor Lite. In: AIED 2009. IOS Press, Amsterdam, The Netherlands (2009)
5. Murray, T.: An overview of intelligent tutoring system authoring tools. In: Authoring Tools for Advanced Technology Learning Environments, pp. 493– 546 (2003)
6. Nye, B.D.: ITS and the digital divide: Trends, challenges, and opportunities. In: AIED 2013 (2013)
7. Nye, B.D., Graesser, A.C., Hu, X.: Multimedia learning in intelligent tutoring systems. In: Mayer, R.E. (ed.) Multimedia Learning (3rd Ed.). Cambridge University Press (2013)
8. Pavlik, P.I., Maass, J., Rus, V., Olney, A.M.: Facilitating co-adaptation of technology and education through the creation of an open-source repository of interoperable code. In: ITS 2012. pp. 677–678. Springer, Berlin (2012)
9. Pavlik, P.I., Toth, J.: How to build bridges between intelligent tutoring system subfields of research. In: Aleven, V and Kay, J and Mostow, J. (ed.) ITS 2010. LNCS, vol. 6095, pp. 103–112 (2010)
10. Peter Wiemer-Hastings, Arthur C. Graesser, Derek Harter: The foundations and architecture of AutoTutor. In: Goettl, B.P., Halff, H.M., Redfield, C.L., Shute, V.J. (eds.) ITS 1998. LNCS, vol. 1452, pp. 334–343. Springer, Berlin (Sep 1998)
11. Raymond, E.S.: The Art of UNIX Programming. Addison-Wesley (2003)
12. Roschelle, J., Kaput, J.: Educational software architecture and systemic impact: The promise of component software. Journal of Educational Computing Research 14(3), 217–228 (1996)
13. Sottilare, R.A., Goldberg, B.S., Brawner, K.W., Holden, H.K.: A modular framework to support the authoring and assessment of adaptive computer-based tutoring systems (CBTS). In: I/ITSEC (2012)
14. VanLehn, K.: The behavior of tutoring systems. International Journal of Artificial Intelligence in Education 16(3), 227–265 (2006)
15. Woolf, B.: Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning (2009)

Authors:

Benjamin D. Nye is a post-doctoral fellow at the University of Memphis, working on tutoring systems architectures as part of the ONR STEM Grand Challenge. Ben received his Ph.D. from the University of Pennsylvania and is interested in ITS architectures, educational technology for development, and cognitive agents.

Leveraging a Generalized Tutoring Framework in Exploratory Simulations of Ill-Defined Domains

James M. Thomas[1], Ajay Divakaran[2], Saad Khan[2]

[1]*Soar Technology, Inc., Ann Arbor, MI*
jim.thomas@soartech.com

[2]*SRI International Sarnoff, Princeton NJ*
{ajay.divakaran, saad.khan}@sri.com

Abstract. Generalized frameworks for constructing intelligent tutors, such as GIFT promise many benefits. These benefits should be extended to systems that work in ill-defined domains, especially in simulation environments. This paper presents ideas for understanding how ill-defined domains change the tutoring dynamic in simulated environments and proposes some initial extensions to GIFT that accommodate these challenges.

Keywords. Intelligent Tutoring Systems, Computer-Based Tutoring Systems, framework, GIFT, ITS, CBTS, student models, learner models, task models, ill-defined domains, simulated environments, exploratory learning.

1 Introduction

Intelligent Tutoring System (ITS) have been shown to enhance learning effectiveness in a wide variety of academic domains [1]. The ITS field has long drawn inspiration from studying strategies employed by human tutors in one-on-one engagement with students [2]. Success has spurred the research community to extend its aspirations into more complex, ill-defined domains. Ill-defined domains are those that lack clearly defined procedures to determine the correctness of proposed solutions to specific problems [3]. Our interest lies in exploratory training simulations of those domains.

To address the difficulty of guiding effective learning in these complex environments, it seems useful to develop and leverage generalized techniques. The GIFT architecture represents a comprehensive approach to facilitate reuse of common tools and methods for building ITS. Although much of the initial focus of GIFT has been directed toward well-defined domains, it we would like to consider how it could be extended to ill-defined domains as well [4], especially those rendered through exploratory simulations.

The authors' motivating example is a system we are building called "Master Trainer – Individualized" (MT-I). The goal for this system is to intelligently guide new military squad leaders in simulations that combine intercultural communication and negotiation skills with tactical challenges. This system integrates stand-off assessments of student affect to modulate the intensity of the simulation to optimize student challenge. One of the questions we are investigating is how to drive the relationship between the student and a simulated human to achieve pedagogically useful levels of anger, conflict or cooperation. We are interested in applying what we are learning toward the generation of useful domain-independent strategies that could be incorporated into GIFT.

2 Motivations for GIFT

Although the field of ITS research is imbued with a strong collaborative spirit, the field lacks common computational infrastructure. GIFT is a particularly promising approach toward a general reusable framework for intelligent tutoring could benefit the entire field.

Scientific research largely presumes the capability to make apples-to-apples comparisons of competing theories. Although they share some common concepts and goals, the majority of ITS research systems share little common architecture or code [1]. To make broad contributions to this field often requires a fairly full-featured ITS on which to perform analyses, yet bespoke software development is both time consuming and expensive. Shared platforms and plug-ins amortize development costs and grow communities of professionals who can more effectively collaborate and relocate between projects and organizations, accelerating the productivity of the field as a whole [5].

GIFT proposes common frameworks for alternative implementations of a broad set of ITS capabilities. Built on solid design principles and a comprehensive understanding of the work of the ITS community, GIFT promises to serve an increasingly useful role in accelerating the scientific and commercial success of the field. Three common challenges faced by the field: authoring, instructional management, and analysis form the core constructs of GIFT. Successful evolution of these constructs promises to accelerate scientific progress by sharing common evaluation methodologies, reducing the time and expense for reused software components, and promoting a more tightly integrated and collaborative community.

GIFT may help accelerate commercialization of scientific progress by facilitating the production of a common currency of evidence of learning effectiveness that can be used to sell the benefits of implemented systems. It can help provide a platform for rapid prototyping to more quickly cycle through alternative approaches to find those that work best. Much as Eclipse™ has accelerated software development [5], and Unity3D™ has democratized high fidelity game development [6], GIFT has the potential to grow into a common workbench that builds-in the ability to package and deploy new work to a full breadth of possible platforms.

3 Characteristics of Ill-Defined Domains

The current GIFT vision accommodates a wide range of ITS capabilities. However, ill-defined domains have not been a primary component of that vision [4]. This section begins with an irony-free definition of ill-defined domains, describes some of the challenges encountered by human tutors in a subset of these domains, and then considers issues and opportunities they present for ITS designers working with immersive simulation environments.

3.1 Defining Ill-Defined Domains

Much of the historical grounding of ITS research is focused on guiding students through well-structured discrete learning tasks, to impart deeply decomposable knowledge [5] from well-defined domains. Fournier-Viger et al. [8] declare ill-defined domains to be those "where traditional approaches for building tutoring systems are not applicable or do not work well". Lynch and Pinkus [9] characterize problems in ill-defined domains as lacking definitive answers, having answers heavily dependent upon the problem's conception, and requiring students to both retrieve relevant concepts and map them to the task at hand. Mitrovic [10] underscores the important distinction between ill-defined domains and ill-defined tasks, anticipating Sottilare's [4] observation that ITS authoring in ill-defined domains is complicated by the multiplicity of "paths to success" compared to the more well-defined domains in which of ITS research has been situated.

3.2 Tutoring Challenges Posed by Ill-Defined Domains

Human tutors have served as both an inspiration for ITS behavior and benchmark and a benchmark for ITS performance [1]. Because no one has yet made a comprehensive study comparing human tutor behaviors in traditional domains with those in ill-defined domains to identify the most necessary extensions to tutorial reasoning, our work on the MT-I system is inspired by specific analogues of human tutors in the domains of live military training for tactics and intercultural effectiveness.

Live training in environments that combine military tactics and interpersonal challenges often spans many hours or days, ranges through confined indoor and expansive outdoor spaces, and requires dozens or hundreds of live role players. Interactions with these role players are often guided by scripted prompts, but involve a lot of improvisation as well. Examples include resolving disputes between armed civilians, negotiating with civic or spiritual leaders, as well neutralizing threats posed by snipers or potential ambushes. Trainers are usually embedded within the environment and have the ability to provide guidance to students during the simulation.

When comparing the behavior of the trainers/tutors in these exercises to that of academic tutors, a striking contrast is immediately evident. Feedback is often deferred over much longer intervals than what one would typically see in one-on-one tutoring in well-defined academic domains. Because is often unsafe or impossible to suspend exercises involving moving/flying vehicles and timed explosions, most incorporate extensive after-action review (AAR) as the primary conduit for feedback and guidance. To some extent, the tutors may elect to integrate feedback within the broader context of a scheduled AAR. In other cases, immediate feedback cannot be given on an individual student action choice because multiple student actions are required before a judgment can be made. Some of this deferral is linked to the interplay between student and role players, as it is difficult to guide the student without impacting the on-going social exchange. Finally, unlike many academic tasks, it is difficult to reset the problem state after an incorrect student action, as the training is situated in a narrative context with a fixed rate of flow to coordinate the many moving parts.

The immediate feedback tutors do provide in these simulations is often constrained to ensuring that the student is engaged and taking actions that move the implicit narrative forward. The deferred feedback is often a holistic reflection involving multiple learning objectives, student affect and metacognitive guidance on productive application of the feedback to future performance.

3.3 Tutoring in Computer Simulations of Ill-Defined Domains

Many of the challenges encountered by humans in ill-defined domains carry over into computer-based tutoring. The assessment granularity sometimes spans multiple actions, can sometimes be entwined in social interactions, and can sometimes be entwined in narrative. Each of these specific constraints can be viewed more generally.

What we commonly describe as narrative in simulation environments is more generally described as a meaningful continuity of state over time. Narrative-centered learning environments [11, 12] can vary in the extent to which they support alternative branches toward "success" or even emergent run-time generation. Yet they share the constraint that the continuity associated with the progression of states cannot be broken or the reversed without consequence, which in turn places limitations on the action choices available to both student and tutor.

Similarly, what we commonly perceive as social interaction between students and non-player characters (NPCs) in simulations is one particular case of an addition of simulation-based elements to tutorial state. In this case, it is the game-state data associated with the NPCs attitudes toward the student that persistent over sequences of tutorial actions. Other examples of game/simulation state variables that influence tutorial state include consumable or non-replenishable resources in the simulation which may affect the span of future tutorial choices.

Finally, the dependency on multiple student actions for student assessment is a specific manifestation of the well-recognized and more general problem of assessing student correctness at all in ill-defined domain. Yet while these challenges complicate the job of intelligent tutoring, they also introduce new tools. Narrative continuity can be exploited both to scaffold instruction and provide context for interpretation of actions. NPCs and other simulation based entities can be manipulated for pedagogical purposes, providing implicit guidance or challenge to the student. The complexity of interpretation of student action affords the intelligent tutor the opportunity for more nuanced and complex forms of guidance that may have more profound and lasting effects on learning.

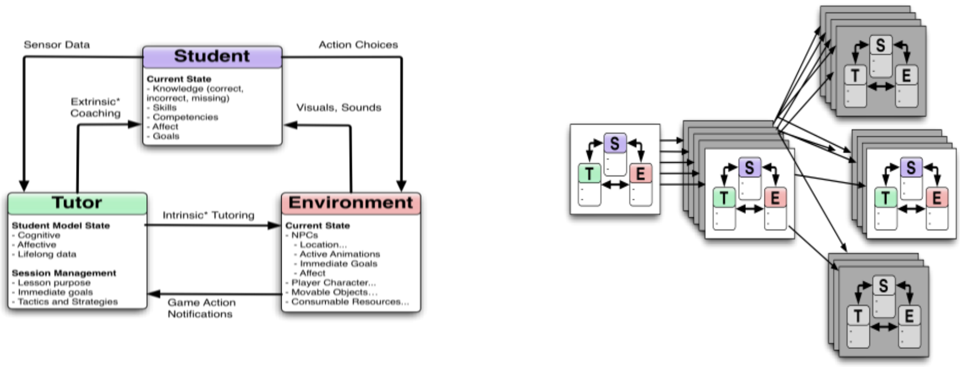


Fig. 8. Model of Tutoring Dynamic in Simulated Ill-Defined Domains

To best confront the challenges and make use of the opportunities of learning based in simulation environments over ill-defined domains, we need models that understand the tutoring dynamic as more than a one-on-one exchange. Rather, the model must recognize that the persistent state, continuity constraints, and assessment ambiguity of the simulation environment continually shape the interactions between tutor and student. Figure 1 is a depiction of such a model. At any point in time, the state of an ITS can be described as a combination of the state data associated with the student, the tutor and the simulation environment. Arrows depict the flow of state-changing actions between these three components. Note that some of these actions may proceed in parallel and may last for human-perceptible durations; perhaps with sufficient frequency that the overall state of the system may be more often in flux than it is quiescent.

This expanded interaction model complicates the prescription of the "learning effect chain" [4]. Because any change to student, tutor, or environment is represented as a new state, the progression toward learning gains involves navigating through a broad space of potential alternative paths. As shown in the rightmost half of Figure 1, one particular progression (the sequence of colored tutorial state snapshots depicted against white backgrounds) is merely one path through a rapidly expanding profusion of alternatives.

This tableau of interwoven learning progressions and alternatives gives an ambitious tutorial agent a lot to think about. Sufficiently inspired agents may perform plan-based reasoning to map the possibilities and nudge the learning experience in the most fruitful directions. In fact, tutorial agents have been constructed that mine the space of alternative actions sequences [12] to devise remediation strategies. Advanced agents might even consider choreographing multiple sessions, altering emphasis and tactics as it varies the pedagogical purpose of each session.

Alternatively, the profusion of possibilities can influence time-sensitive developers to move in the other direction, building "knowledge-lean" [13tutorial agents. As a consequence, ITS developers in these environments often eschew deep knowledge-tracing expert models in favor of less precise, but more easily authored constraint-based approaches [8]. This suggests that a generalized intelligent framework, such as

GIFT, should consider supporting a variety of modeling approaches. In fact, our current implementation of MT-I, which features ill-defined tasks within overlapping ill-defined domains, we have found it useful to author constraint-based models to characterize the correctness of individual student tasks in a wide range of potential contexts, where that model feeds a higher-level knowledge tracing model of higher-level, more abstract learning objectives that operates over longer time spans.

4 Enhanced Knowledge Representations and Reasoning

Not surprisingly, some of the challenges posed by ill-defined domains in simulated environments can be addressed by providing tools to create better definitions. Flexible and knowledge representations (KR) can serve as the definitional "handles" that tutorial agents can use to enhance reasoning about the state of the student and simulation. That reasoning can be converted to action if the simulation is instrumented with "levers", software hooks that cause pedagogically useful changes expressed through those handles. This section proposes three levers that use non-traditional extensions to tutorial knowledge representations to provide enhance tutorial reasoning and more effective student guidance.

Lever #1: Enriched Definitions of Learning Objectives. Trainers in the sophisticated simulations involving role players discussed earlier are often trying to steer their students toward states of mind that go beyond a prescribed set of factual knowledge to include social, narrative and affective dimensions, as shown in Figures 1 and 2. To achieve similar results in simulated environments, tutorial agents must reason about those dimensions of learning objectives as well. The KR should be able to qualify, for example, not only that the trainee know how to greet respectfully a village leader, but that the student can perform that greeting is accomplished while in a highly agitated state.

Lever #2: Enriched Definitions of Tutorial Purpose. Sophisticated simulations can be used in a broader set of pedagogical contexts that traditional systems, ranging from direct instruction of material to which the student has not previously been exposed, to consolidation of previously taught material, to transfer of knowledge to new domains, to assessment of knowledge and performance, to building confidence, teamwork or skills that apply acquired knowledge. Thus, the purpose of a given tutorial session can vary more widely than in traditional instruction, which demands that tutorial strategies and tactics be labeled according to their relevance for these various purposes. For example, a particular tutorial action may have a stronger positive effect on student self-efficacy that an alternative which may have a stronger positive effect on didactic specificity. An enhanced KR enables the tutorial agent to choose between these alternatives based on whether the purpose of the current session is to build confidence or impart knowledge.

Lever #3: Persistently-labeled Learner Data. To maximally leverage the opportunities of sophisticated learning environments, in which multiple learning sessions for varying learning purposes may span arbitrary time periods, individual student data must be persistent and pervasive: accessible and publishable at any level by any component of the tutorial framework. This allows tutorial agents running at various levels with various time horizons to tie together data collected on individual students across multiple sessions. For example, it could prove useful to know how quick a student is to anger, or which immediately reachable emotional states are most conducive to learning for a particular student, where that data may have been collected and stored in an earlier tutorial session by an agent using the same generalize frame work. All student model data should be tagged with its expected lifespan: step, task, session, application, or beyond. This enhances the ability of any particular tutorial agent to perform macro-level adaptation [14] to evolve learning across multiple domains that enhance domain-independent competencies.

5 Conclusions

A generalized framework like GIFT holds significant promise for accelerating scientific and commercial success of ITS. Yet one of the areas in which that acceleration is most desperately needed, ill-defined domains in simulated environments, are not addressed in depth by the current approach to GIFT. We suggest that a first step in this direction would to explore several extensions to the knowledge representations in GIFT to meet the demands of those environments.

6 Acknowledgements

This research effort was supported in part by contract W911QX-12-C-0149 from the US Army Research Laboratory. We gratefully acknowledge the interest and encouragement from Dr. Bob Sottilare at ARL. In addition, Scott Flanagan of Sophia Speira, LLC, has provided invaluable understanding of military training practices and domains.

7 References

1. VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, *46*(4), 197-221.
2. Wood, D., Bruner, J. S., & Ross, G. (1976). The role of tutoring in problem solving*. *Journal of child psychology and psychiatry*, *17*(2), 89-100.
3. Minsky, M. (1995). Steps to artificial intelligence. In Luger, G. F., editor, *Computation and Intelligence, Collected Readings*, chapter 3, pages 47–90. AAAI, Menlo Park CA, and MIT Press.
4. Sottilare, B. (2012). Considerations in the development of ontology for a generalized intelligent framework for tutoring. *Proceedings of the International Defense and Homeland Security Simulation Workshop*.
5. Kidane, Yared H., and Peter A. Gloor. "Correlating temporal communication patterns of the Eclipse open source community with performance and creativity." *Computational and mathematical organization theory* 13.1 (2007): 17-27.
6. Torrente, Javier, et al. "Game-like simulations for online adaptive learning: A case study." *Learning by Playing. Game-based Education System Design and Development*. Springer Berlin Heidelberg, 2009. 162-173.VanLehn, K. (1990). Mind bugs: the origins of procedural misconception. MIT press.

7. VanLehn, K. (1990). Mind bugs: the origins of procedural misconception. MIT press.
8. Fournier-Viger, P., Nkambou, R., & Nguifo, E. M. (2010). Building Intelligent Tutoring Systems for Ill-Defined Domains. In *Advances in Intelligent Tutoring Systems* (pp. 81-101). Springer Berlin Heidelberg.
9. Lynch, C., Ashley, K., Aleven, V., & Pinkwart, N. (2006). Defining ill-defined domains; a literature survey. In Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains at the 8th International Conference on Intelligent Tutoring Systems (pp. 1-10).
10. Mitrovic, A., & Weerasinghe, A. (2009). Revisiting Ill-Definedness and the Consequences for ITSs. Artificial Intelligence in Education: Building Learning Systems That Care: from Knowledge Representation to Affective Modelling, 200, 375.
11. Mott, B., McQuiggan, S., Lee, S., Lee, S. Y., & Lester, J. C. (2006). Narrative-centered environments for guided exploratory learning. In *Proceedings of the AAMAS 2006 Workshop on Agent-Based Systems for Human Learning* (pp. 22-28).
12. Thomas, J. M., & Young, R. M. (2009, July). Using Task-Based Modeling to Generate Scaffolding in Narrative-Guided Exploratory Learning Environments. In *Proceeding of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modeling* (pp. 107-114).
13. Lane, H. C., Core, M. G., Gomboc, D., Karnavat, A., & Rosenberg, M. (2007, January). Intelligent tutoring for interpersonal and intercultural skills. In *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*(Vol. 2007, No. 1). National Training Systems Association.
14. Sottilare, R. A., Goldberg, B. S., Brawner, K. W., & Holden, H. K. (2012, January). A Modular Framework to Support the Authoring and Assessment of Adaptive Computer-Based Tutoring Systems (CBTS). In *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*(Vol. 2012, No. 1). National Training Systems Association.

Authors

James M. Thomas: James M. (Jim) Thomas is a Research Scientist at Soar Technology. His current work includes intelligent training and assessment systems that aid children on the autism spectrum and guide soldiers to integrate socio-cultural and tactical skills. He received his Ph.D.degree in Computer Science from North Carolina State University in 2011. His dissertation entitled “Automated Scaffolding of Task-Based Learning in Non-Linear Game Environments”, explored general mechanisms to generate intelligent tutorial planning within exploratory learning environments. Jim has authored more than a dozen papers in the area of intelligent tutoring systems, automated planning, and computer games. His graduate studies were supported by a National Science Foundation Graduate Research Fellowship in Artificial Intelligence. Concurrent with his doctoral studies, Jim developed game-based learning systems designed to improve children’s social skills at the 3-C Institute for Social Development, including work which was published in the journal *Child Development*. Jim also benefits from 15 years of experience in the computer and telecommunications industries, including software development, management, and senior marketing management with IBM, BNR and Nortel Networks.

Ajay Divakaran: Ajay Divakaran, PhD is a Technical Manager at SRI International Sarnoff. He has developed several innovative technologies for multimodal systems for both commercial and government programs over the past 16 years. He currently leads SRI Sarnoff’s projects on Modeling and Analysis of Human Behavior for the DARPA SSIM project, ONR Stress Resiliency project, Army "Master Trainer" Intelligent Tutoring project among others. He worked at Mitsubishi Electric Research Labs for ten years where he was the lead inventor of the world's first sports highlights playback enabled DVR, as well as a manager overseeing a wide variety of product applications of machine learning. He was elevated to Fellow of the IEEE in 2011 for his contributions to multimedia content analysis. He established a sound experimental and theoretical framework for human perception of action in video sequences, as lead-inventor of the MPEG-7 video standard motion activity descriptor. He serves on TPC's of key multimedia conferences and served as an associate editor of the IEEE transactions on Multimedia from 2007 to 2011 and has two books and over 100 publications to his credit as well as over 40 issued patents. He received his Ph.D. degree in Electrical Engineering from Rensselaer Polytechnic Institute in 1993.

Saad Khan: Saad Khan is a Senior Computer Scientist at SRI International with 10 years of experience developing computer vision algorithms. He has led the design and development of advanced military training systems that can adapt to both training scenarios and learners' behavior. He serves as PI/ Co-PI and technical lead on programs in multimodal sensing algorithms for immersive training for DARPA, ONR and PMTRASYS. He led the development and transition of APELL (Automated Performance Evaluation and Lessons Learned) training system. APELL is an immersive, interactive, Mixed Reality training system that has been successfully deployed at the Marines Camp Pendleton training facility. Prior to joining SRI Sarnoff, Dr. Khan conducted research on 3D model based object tracking and human activity analysis in the IARPA VACE program. His work in automated image based localization earned an Honorable Mention award at the International Conference of Computer Vision 2005. He has authored over 20 papers and has 2 issued patents. His work on multiple view human tracking is one of the most highly cited works in recent tracking literature. He received his PhD in Computer Science from University of Central Florida in 2008.

Toward "Hyper-Personalized" Cognitive Tutors

Non-Cognitive Personalization in the Generalized Intelligent Framework for Tutoring

Stephen E. Fancsali[1], Steven Ritter[1], John Stamper[2], Tristan Nixon[1]

[1]*Carnegie Learning, Inc.*
437 Grant Street, Suite 918
Pittsburgh, PA 15219, USA
{sfancsali, sritter, tnixon}@carnegielearning.com

[2]*Human-Computer Interaction Institute*
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, USA
john@stamper.org

Abstract. We are starting to integrate Carnegie Learning's Cognitive Tutor (CT) into the Army Research Laboratory's Generalized Intelligent Framework for Tutoring (GIFT), with the aim of extending the tutoring systems to understand the impact of integrating non-cognitive factors into our tutoring. As part of this integration, we focus on ways in which non-cognitive factors can be assessed, measured, and/or "detected." This research provides the groundwork for an Office of the Secretary of Defense (OSD) Advanced Distributed Learning (ADL)-funded project on developing a "Hyper-Personalized" Intelligent Tutor (HPIT). We discuss the integration of the HPIT project with GIFT, highlighting several important questions that such integration raises for the GIFT architecture and explore several possible resolutions.

Keywords: Cognitive Tutors, intelligent tutoring systems, student modeling, affect, personalization, non-cognitive factors, gaming the system, off-task behavior, Generalized Intelligent Framework for Tutoring, GIFT

1 Introduction

Our goal in developing a "Hyper-Personalized" Intelligent Tutor (HPIT) is to bring learning systems to the next level of user/student adaptation. In addition to traditional features of systems like Carnegie Learning's Cognitive Tutor (CT), HPIT includes non-cognitive factors to provide a more personalized experience for users of the system. In this paper, we discuss features of HPIT and situate the work in the context of the Generalized Intelligent Framework for Tutoring (GIFT) architecture.

1.1 Cognitive Tutors

Carnegie Learning's Cognitive Tutor (CT) [1] is an adaptive, computer-based tutoring system (CBTS) or intelligent tutoring system (ITS) based on the Adaptive Control of Thought—Rational (ACT-R) theory of cognition [2] used every year by hundreds of thousands of learners, from middle school students through college undergraduates. To date, Carnegie Learning's development of the CT has focused primarily on mathematics.

1.2 Generalized Intelligent Framework for Tutoring (GIFT)

The Army Research Laboratory (ARL) is working to develop the Generalized Intelligent Framework for Tutoring (GIFT). The GIFT project aims to provide a "modular CBTS framework and standards [that] could enhance reuse, support authoring and optimization of CBTS strategies for learning, and lower the cost and skillset needed for users to adopt CBTS solutions for military training and education" [3]. Given substantial efforts in both academia and industry to develop ITSs, integrating aspects of this work with ARL's GIFT is important for future development. We briefly provide an overview of GIFT before describing a particular project that will integrate architecture for "hyper-personalized" versions of ITSs, like Carnegie Learning's CT, with GIFT.

GIFT provides a modular framework to achieve and support three goals or "constructs" [3]: (1) affordable, easy authoring of CBTS components, (2) instructional management for integrating pedagogical best practices, and (3) experimental analysis of effectiveness.

GIFT's service-oriented architecture (SOA) currently provides four modules, among other functional elements, around which CBTSs can be implemented and into which existing ITSs can be integrated. Three modules are domain-independent: the *Sensor Module*, *User Module*, and *Pedagogical Module*. The *Domain Module* contains all domain-specific content, including problems sets, hints, misconceptions, etc.

One functional element outside of "local tutoring processes" in the GIFT architecture is important for the present discussion: *Persistent Learner Models*. These models are intended to "maintain a long term view of the learner's states, traits, demographics, preference, and historical data (e.g., survey results, performance, competencies)" [3]. As we review several key, non-cognitive factors upon which we seek to base a "hyper-personalized" CT, the importance of data intended to be tracked by *Persistent Learner Models* will be clear. However, the notion of "persistence" for this data becomes less clear.

2 Non-Cognitive Factors

While the CT and other ITSs adapt content presented to students based on cognitive factors such as skill mastery, there are many other (cognitive and non-cognitive) factors for which the student learning experience might be adapted and personalized. We present several examples of recent research focusing on the impact of non-cognitive factors on student learning in ITSs.

2.1 Gaming the System and Off-Task Behavior

A wide variety of behaviors in an ITS or CBTS like CT may be associated with learning. Two specific behaviors that have been widely studied in the recent literature include "gaming the system" behavior and off-task behavior [4] [5]. This research has not only studied the association of these behaviors with learning but has also led to the development of software "detectors" of such behavior from ITS log data.

Sometimes students attempt to advance through material in ITSs like the CT without actually learning the content and developing mastery of appropriate skills. Such behavior is generally referred to as "gaming the system." Examples of such behavior include rapid or systematic guessing and "hint-abuse." "Hint-abuse" refers to repeated student hint requests, sometimes until a final or "bottom-out" hint (essentially) provides the answer to a problem or problem step [10].

Software "detectors" of gaming the system behavior have been developed (e.g., [7]) and correlated with field-observations of student behavior. Such software detectors rely on various features that are "distilled" from CT log files [8]. Studies find an association [4] [9] [10] and evidence for a causal link [11] [12] between gaming the system behavior and decreased student learning. Similar software has also been developed, and validated via field-observations, to detect off-task behavior [5].

Other types of behavior, of course, may also be important for learning in CBTSs and ITSs. While some behaviors may be "sensed" via physical, tactile, and/or physiological sensors, we emphasize that state-of-the-art research attempts to detect different types of behavior from logs generated by CBTSs and ITSs.

2.2 Affect

Building on success in developing detectors of student behavior, current research seeks to detect student affect (e.g., boredom, engaged concentration, frustration, etc.) without sensors (i.e., without physical, tactile, and/or physiological sensors) [13]. Such detectors have also been validated by field-observations of students using ITS in the classroom. Further, these detectors have been successfully deployed to predict student learning via standardized test scores [14].

While student affect and behavior might also be physically "sensed", inferred, or measured via survey instruments (e.g., mood via survey [15]), data-driven detection of student affect and behavior is a promising approach to achieve the GIFT design goal of supporting "low-cost, unobtrusive (passive) methods… to inform models to classify (in near real-time) the learner's states (e.g., cognitive and affective)" [3].

2.3 Preferences

Carnegie Learning's middle school mathematics CT product, MATHia, allows students to set preferences for various interest areas (e.g., sports, art) and probabilistically tailors problem presentation based on those preferences. On-going research aims to determine if and how presenting students with problems related to their preference areas is associated with engagement and benefits student learning (e.g., [16-17]). Other student preferences might be ascertained via surveys, configuration settings, or inferred from data, at different levels of granularity and time scales.

2.4 Personality and Other Learner Characteristics

Other characteristics of learners may prove important for learning. We consider two prominent examples that are being considered as we develop HPIT. Investigating other learner characteristics is also a topic for future research.

Grit.

Grit [18-19] is defined as the tendency to persist in tasks over the long term, when reaching the goal is far off in the future. Duckworth et al. [18] found that grit, measured by a survey instrument [19], predicted retention among cadets at the United States Military Academy at West Point, educational attainment among adults, and advancement to the final round among contestants in the Scripps National Spelling Bee.

Educational environments like CT are able to adjust the rate at which difficulty of activities increases. Students high in grit may, for example, benefit more from rapid increases in the difficulty of course material compared to students low in grit, regardless of knowledge-levels.

Motivation and Goals.

Students' motivation and goals are likely to be important for learner adaptation. Recent research [20] considers fine-grained assessment, via frequent surveys (occurring every few minutes) embedded within CT, of student motivation and goal orientation to better understand models self-regulated learning. Elliot's framework for achievement goals provides for two dimensions, definition (mastery vs. performance) and valence (approach vs. avoidance), along which goals are oriented [20-22].

Particular problems or hints (or ways of providing hints) might, for example, be best suited to students with a mastery avoidance goal orientation that seek to avoid failure, and so on. In addition to ascertaining the influence of goals and motivation on learning, determining whether students' motivation and goals (at various levels of granularity) are relatively static or dynamic through a course, and possibly influenced by students' experience in a course, remains a topic of active research [20].

3 Hyper-Personalizing Cognitive Tutors

One particularly important aspect of CTs from an architectural perspective is that they are driven by user inputs (called "transactions" [23]). From a system perspective, an update to the learner model happens only when the student takes some action within the system (e.g., attempting to answer a question or asking for a hint). Other student-initiated inputs might include, for example, student ratings of whether particular problems are interesting (e.g., an ever-present 5-star ranking system attached to each problem). Student-initiated inputs range in time from the nearly continuous to being separated by significant amounts of time.

In a more general system like GIFT, updates to the student model happen, not only at different timescales, but can also be initiated by actors (or factors) other than the learner. Examples include: acquiring data to update the student model through surveys given to the student at times determined by the system (e.g., only at course-beginning and end vs. periodically between problems or units), through real-time sensors (e.g., an eye-tracker), through student-determined inputs, etc. Furthermore, in some learning environments, the student model might be updated by factors linked to the passage of time (e.g., inferring that a skill has been “forgotten” because the student does not use a tutor for a substantial amount of time or updating students’ knowledge state after a chemical reaction occurs following some time-lapse in a simulation-based chemistry tutor). The mode and frequency of data collection, in part, determine the kinds of pedagogical moves that the ITS can take.

The ADL-funded Hyper-Personalized Intelligent Tutor (HPIT) project seeks to develop a modular, plug-in-like architecture using various data collection and processing elements to inform CT’s provision of problems, feedback, hints, etc. Each factor (whether cognitive or non-cognitive) may contribute to varying degrees to the decision-making process, as data are collected and inferences drawn about learner “state.” A plug-in architecture allows for “voting” schemes to drive the personalization process (e.g., perhaps two non-cognitive factors and one cognitive factor are all equally weighted, or not). Methods will be developed to resolve conflicts (i.e., break “ties”) when multiple recommendations are appropriate given a student’s “state.”

While cognitive factors are crucial for adapting educational content for disparate users, HPIT’s primary innovation is the creation of a platform and framework for adapting content based on non-cognitive factors. To do so, HPIT will draw on data from software detectors, surveys, and possibly physical sensors. Perhaps more important from an architectural perspective, however, is the fact that the measurement, inference, or assessment of various cognitive and non-cognitive factors may occur on different time scales and at different levels of granularity.

For example, if a student is both bored (as, for example, inferred from a software detector applied to real-time log data) and uninterested in material currently being presented (as inferred from survey results), material similar in difficulty, but providing examples better suited to student preferences, might be presented. However, a different strategy might be required if we lack data about their interests. Adapting pedagogical strategies based on data that is currently available is a virtue of the flexibility of the HPIT architecture we are developing.

4 Implications for GIFT Architecture

The GIFT architecture and recent research (e.g., [15]) focuses on using physical sensors and surveys to gather information about a learner’s non-cognitive state. The HPIT framework builds on work to infer/measure student state with surveys and software detectors that use data from tutor logs. These software detectors rely on data generated by the ITS following student-initiated input to the ITS. We discuss several implications for the GIFT architecture and the integration of existing ITSs into GIFT.

4.1 Surveys

In GIFT, *Persistent Learner Models* store survey results and communicate with the *User/Learner Module* via the SOA. However, HPIT requires that surveys be deployable at nearly any point in the learning experience, rather than simply before and after a "chunk" (e.g., unit) of course material. Furthermore, surveys/polls might be conducted that assess momentary characteristics of the student experience, rather than the persistent state of a student[6].

Some survey-like elements may be deployed nearly continuously. Thus, it might be initially attractive to conceptualize surveys are as a particular type of sensor. However, the processing of the type of survey data we have in mind seems fundamentally different than processing sensor data (e.g., an eye-tracker). Consider, for example, the previously noted five-star rating system for problems. While the rating system may be deployed for near-continuous collection of data, frequently students may not choose to rate many problems. Perhaps we find that a student who rates problems infrequently assigns two particular problems a 1-star (low) rating. Given the lack of input from this student, these data may be especially salient and require special consideration compared to a student who frequently rates problems, and with high variability. Such possibilities seem to suggest that we treat discrete survey data (even with high-polling rates) differently than sensors that continuously provide data.

4.2 (Sensor-Free) Detectors in the GIFT Architecture

For purposes of software implementation, detectors are essentially sensors (i.e., both process, filter, and/or aggregate streams of data to make inferences about student state); "detector processing" would be nearly identical to "sensor processing" within the *Sensor Module*. However, the input characteristics of software detectors are much different than those of sensors in the GIFT architecture, as the notion of a sensor within GIFT, to date, focuses on physical sensors. Detectors generally rely on student/user-initiated input mediated by the learning environment, but detectors might also be developed that do not rely on user-initiated input (e.g., a detector of "forgetting" based on time-lapse in usage of the ITS).

One possible resolution would have the *Domain Module* (and/or the *Tutor-User Interface*) as input to the *Sensors* element, so that software-based detectors that rely on tutor log data are also conceptualized as *Sensors*. This proposal may stretch the notion of *Sensors* too far. In response, one might include a new type of *Detector/Analysis Module* that would take *Domain Module* (and possibly *Pedagogical Module* or *Tutor-User Interface*) data as input and provide information to the *User Module* about learners' affective and cognitive states via software detectors. This achieves the goal of keeping the relatively domain-independent detectors outside of the *Domain Module*. This requires that *Domain Module* output is sufficiently rich for use by detectors; as currently conceptualized, this is not clear.

[6] The HPIT architecture maintains such flexibility so that the investigator is free to make (or not make) distinctions about persistent versus non-persistent student characteristics (and concomitant timing decisions about assessment, measurement, or detection).

5 Discussion

Overall, we suggest that the GIFT architecture is well-served by considering the consequences of integrating a broader range of input and output relationships among its component modules (or possibly new types of modules) and other functional elements, including considerations of the presence, timing, granularity, and content of data passed between components.

Current research provides for data-driven means to use CT (and other CBTS) logs to classify and "detect" student behavior and affect without physical sensors, whether transactions and inputs are student-initiated or system-initiated. Integrating capabilities necessary for HPIT will be a fruitful extension of GIFT.

Furthermore, detectors rely on data from the ITS to determine whether students are off-task, gaming, bored, frustrated, etc. Such detectors require relatively rich log data and would not be served by the impoverished (i.e., abstract) assessment categories of "above standard," "below standard," etc., provided by the *Domain Module*. This suggests that detectors are a part of the *Domain Module*, but they are also (relatively) domain independent. Thus, it is not clear that they should be included in the *Domain Module*. Requiring detectors be a part of the *Domain Module* would also incur costs in terms of reusability and modularity. Alternatively, richer data might be provided to an enhanced *Learner Module* that subsumes (aspects of) the *Sensor Module* and our proposed detectors (i.e., the *Detector/Analysis Module*) to better infer characteristics of a learner's state. Further, other open questions remain as to the proper placement of other components of CTs within the GIFT architecture.

6 References

1. Ritter, S., Anderson, J.R., Koedinger, K.R., Corbett, A.T.: Cognitive Tutor: Applied Research in Mathematics Education. Psychonomic Bulleting & Review 14, 249-255 (2007)
2. Anderson, J.R.: Rules of the Mind. Erlbaum, Hillsdale, NJ (1993)
3. Sottilare, R.A., Brawner, K.W., Goldberg, B.S., Holden, H.K.: The Generalized Intelligent Framework for Tutoring (GIFT). (2012), http://www.gifttutoring.org/
4. Baker, R. S., Corbett, A. T., Koedinger, K .R., & Wagner, A. Z.: Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game the System". In: Proceedings of ACM CHI 2004: Computer-Human Interaction, pp. 383-390 (2004)
5. Baker, R.S.J.d.: Modeling and Understanding Students' Off- Task Behavior in Intelligent Tutoring Systems. In: Proceedings of the 2007 Conference on Human Factors in Computing Systems, pp. 1059-1068 (2007)
6. Aleven, V., & Koedinger, K. R.: Limitations of Student Control: Do Students Know When They Need Help? In: Proceedings of the 5th International Conference on Intelligent Tutoring Systems, pp. 292-303 (2000)
7. Baker, R.S.J.d., de Carvalho, A. M. J. A.: Labeling Student Behavior Faster and More Precisely with Text Replays. In: Proceedings of the 1st International Conference on Educational Data Mining, pp. 38-47 (2008)
8. Baker, R.S.J.d., Corbett, A.T., Roll, I., Koedinger, K.R.: Developing a Generalizable Detector of When Students Game the System. User Modeling & User-Adapted Interaction 18, 287-314 (2008)

9. Walonoski, J.A., Heffernan, N.T.: Detection and Analysis of Off-Task Behavior in Intelligent Tutoring Systems. In: Proceedings of the 8th International Conference on Intelligent Tutoring Systems, pp. 382-391 (2006)
10. Cocea, M., Hershkovitz, A., Baker, R.S.J.d.: The Impact of Off-Task and Gaming Behavior on Learning: Immediate or Aggregate? In: Proceedings of the 14th International Conference on Artificial Intelligence in Education, pp. 507-514 (2009)
11. Fancsali, S.E.: Variable Construction and Causal Discovery for Cognitive Tutor Log Data: Initial Results. In: Proceedings of the Fifth International Conference on Educational Data Mining, pp. 238-239 (2012)
12. Fancsali, S.E.: Constructing Variables that Support Causal Inference. Ph.D. Thesis, Department of Philosophy, Carnegie Mellon University, Pittsburgh, PA, USA (2013)
13. Baker, R.S.J.d., Gowda, S.M., Wixon, M., Kalka, J., Wagner, A.Z., Salvi, A., Aleven, V., Kusbit, G., Ocumpaugh, J., Rossi, L.: Sensor-free Automated Detection of Affect in a Cognitive Tutor for Algebra. In: Proceedings of the 5th International Conference on Educational Data Mining, pp. 126-133 (2012)
14. Pardos, Z.A., Baker, R.S.J.d., San Pedro, M.O.C.Z., Gowda, S.M., Gowda, S.M.: Affective States and State Tests: Investigating How Affect Throughout the School Year Predicts End of Year Learning Outcomes. In: Proceedings of the 3rd International Conference on Learning Analytics and Knowledge (2013)
15. Sottilare, R., Proctor, M.: Passively Classifying Student Mood and Performance within Intelligent Tutors. Educational Technology & Society 15, 101-114 (2012)
16. Walkington, C., Sherman, M.: Using Adaptive Learning Technologies to Personalize Instruction: The Impact of Interest-Based Scenarios on Performance in Algebra. In: Proceedings of the 10th International Conference of the Learning Sciences, pp. 80-87 (2012)
17. Walkington, C.: Using Learning Technologies to Personalize Instruction to Student Interests: The Impact of Relevant Contexts on Performance and Learning Outcomes. Journal of Educational Psychology (forthcoming)
18. Duckworth, A.L., Peterson, C., Matthews, M.D., Kelly, D.R.: Grit: Perseverance and Passion for Long-Term Goals. Journal of Personality and Social Psychology 92, 1087-1101 (2007)
19. Duckworth, A.L., Quinn, P.D.: Development and Validation of the Short Grit Scales (Grit-S). Journal of Personality Assessment 91, 166-174 (2009)
20. Bernacki, M. L., Nokes-Malach, T.J., Aleven, V.: Fine-Grained Assessment of Motivation Over Long Periods of Learning with an Intelligent Tutoring System: Methodology, Advantages, and Preliminary Results. In: Azevedo, R., Aleven, V. (eds.) International Handbook of Metacognition and Learning Technologies. Berlin: Springer (forthcoming)
21. Elliot, A. J., & McGregor, H. A.: A 2 X 2 Achievement Goal Framework. Journal of Personality and Social Psychology 80, 501-519 (2001)
22. Elliot, A. J., & Murayama, K.: On the Measurement of Achievement Goals: Critique, Illustration, and Application. Journal of Educational Psychology 100, 613-628 (2008)
23. Koedinger, K.R., Baker, R.S.J.d., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J.: A Data Repository for the EDM Community: The PSLC DataShop. In: Romero, C., Ventura, S., Pechenizkiy, M., Baker, R.S.J.d. (eds.) Handbook of Educational Data Mining, pp. 43-55 Boca Raton, FL: CRC Press (2011)

Authors

Stephen E. Fancsali is a Research Scientist at Carnegie Learning, Inc. He received a Ph.D. in Logic, Computation, and Methodology from the Philosophy Department at Carnegie Mellon University in May 2013. His doctoral research centered on the construction of variables from fine-grained data (e.g., intelligent tutoring system log files) to support causal inference and discovery from observational data sets. At Carnegie Learning, he focuses on a variety of issues in educational data mining, including student modeling, providing interpretable ways to quantify improvements in cognitive models and other tutoring system components, and statistical and causal modeling of student affect, behavior, and other phenomena as they relate to learning and other education outcomes, especially in intelligent tutoring systems.

Steven Ritter is Co-Founder and Chief Scientist at Carnegie Learning, Inc. He received a Ph.D. in Psychology from Carnegie Mellon University.

John Stamper is Technical Director of the Pittsburgh Science of Learning Center (PSLC) and a faculty member at the Human-Computer Interaction Institute at Carnegie Mellon University. He received a Ph.D. in Information Technology from the University of North Carolina at Charlotte.

Tristan Nixon is a Research Programmer at Carnegie Learning, Inc. He earned a B.S. in Computer Science from the University of Toronto.

Using GIFT to Support an Empirical Study on the Impact of the Self-Reference Effect on Learning

Anne M. Sinatra, Ph.D.

Army Research Laboratory/Oak Ridge Associated Universities
anne.m.sinatra.ctr@us.army.mil

Abstract. A study is reported in which participants gained experience with deductive reasoning and learned how to complete logic grid puzzles through a computerized tutorial. The names included in the clues and content of the puzzle varied by condition. The names present throughout the learning experience were either the participant's own name, and the names of two friends; the names of characters from a popular movie/book series (*Harry Potter*); or names that were expected to have no relationship to the individual participant (which served as a baseline). The experiment was administered using the Generalized Intelligent Framework for Tutoring (GIFT). GIFT was used to provide surveys, open the experimental programs in PowerPoint, open external web-sites, synchronize a Q-sensor, and extract experimental data. The current paper details the study that was conducted, discusses the benefits of using GIFT, and offers recommendations for future improvements to GIFT.

1 Introduction

The Generalized Intelligent Framework for Tutoring (GIFT) provides an efficient and cost effective way to run a study (Sottilare, Brawner, Goldberg, & Holden, 2012). In Psychology research, in-person experiments usually require the effort of research assistants who engage in opening and closing computer windows and guiding participants through the experimental session. GIFT provides an opportunity to automate this process, and requires a minimal knowledge of programming, which makes it an ideal tool for students and researchers in the field of Psychology. GIFT was utilized in the current pilot study, which is investigating the impact of the self-reference effect on learning to use deductive reasoning to solve logic grid puzzles.

1.1 The Self-Reference Effect and Tutoring

Thinking of the self in relation to a topic can have a positive impact on learning and retention. This finding has been consistently found in Cognitive Psychology research, and is known as the self-reference effect (Symons & Johnson, 1997). In addition, research has suggested that linking information to a popular fictional character (e.g., *Harry Potter*) can also draw an individual's attention when they are engaged in a difficult task, and can result in similar benefits to the self-reference effect (Lombardo, Barnes, Wheelwright, & Baron-Cohen, 2007; Sinatra, Sims, Najle, & Chin, 2011). The self-reference effect could potentially be utilized to provide benefits in tutoring and learning. Moreno and Mayer (2000) examined the impact of a participant being taught science lessons in a manner consistent with first person speech (self-reference), or in the third person. No difference was found in regard to knowledge gained from the lessons, however, when asked to apply the knowledge in a new and creative way, those that received the first person instruction demonstrated better performance. This suggests that relating information to the self may result in a "deeper" learning or understanding, which allows the individual to easily apply the information in new situations.

It has been suggested that deep learning should be a goal in current instruction (Chow, 2010). This is consistent with findings that including topics of interest (e.g., familiar foods, names of friends) when teaching math can have a positive impact on learning outcomes (Anand & Ross, 1987; Ku & Sullivan, 2002). Many of the domains (e.g., math, science) that have been examined in the literature are "well-defined" and do not transfer skills to additional tasks. There has not been a focus on deductive reasoning or teaching logic, which is a highly transferable skill. Logic grid puzzles are useful learning tools because they allow an individual to practice deductive reasoning by solving the puzzle. In these puzzles, individuals are provided with clues, a grid, and a story. The story sets up the puzzle, the clues provide information that assists the individual in narrowing down or deducing the correct answers and the grid provides a work space to figure out the puzzle. It has been suggested that these puzzles can be helpful in instruction, as they require the individual to think deeply about the clues and have a full understanding of them in order to solve the puzzle (McDonald, 2007). After practicing deductive reasoning with these puzzles, the skill can then potentially be transferred and applied in many other domains and subject areas.

1.2 The Current Study

The current study sets out to examine the self-reference effect in the domain of deductive reasoning, by teaching individuals how to complete logic grid puzzles. It is a pilot study, which will later be developed into a large scale study. During the learning phase of the study, there were three different conditions: Self-Reference, Popular Culture, and Generic. The study was administered on a computer using GIFT 2.5. The interactive logic puzzle tutorial was developed using Microsoft PowerPoint 2007 and Visual Basic for Applications (VBA). In the Self-Reference condition, participants entered their own name and the names of two of their close friends into the program, in the Popular Culture condition, the participant was instructed to enter names from the *Harry Potter* series ("Harry", "Ron", and "Hermione") into the program, in the Generic condition, participants were instructed to enter names which were not expected to be their own ("Colby", "Russell", and "Denise") into the program. The program then used the entered names throughout the tutorial as part of the clues and the puzzle with which the participants were being taught. Therefore, the participants were actively working with the names throughout their time learning the skill.

After completing the tutorial, participants were asked to recall anything that they could about the content of the puzzle, answer multiple-choice questions about what they learned, answer applied clue questions in which they were asked to draw conclusions based on a story and an individual clue, and complete two additional logic puzzles (one at the same difficulty level as the one in the tutorial, and one more difficult). These different assessments allowed for measures of retention of the learned content, ability to apply the knowledge, and ability to transfer/apply the knowledge in a new situation.

It was hypothesized that there would be a pattern of results such that individuals in the Self-Reference condition would perform better on all assessments than that in the Popular Culture and Generic conditions, and that the Popular Culture condition would perform better on all assessments than the Generic condition. It was also expected that ratings of self-efficacy and logic grid puzzle experience would increase after the tutorial.

1.3 GIFT and the Current Study

The current study required participants to use a computer, and answer survey questions before and after PowerPoint Tutorials and PowerPoint logic grid puzzles. Due to the capabilities of GIFT 2.5 to provide survey authoring and administering, it was an ideal choice for the development of the study. As GIFT has the capability of opening and closing programs (such as PowerPoint), and presenting surveys and instructions in specific orders, it is a highly efficient way to guide participants through a learning environment and a study, without much effort from research assistants.

In Psychology research there are often many different surveys that are administered to participants. An advantage of GIFT is that the Survey Authoring System provides a free and easy to use tool in which to create surveys. A further advantage is that it does not require the individual to be online when answering the survey.

2 Method

2.1 Participants

In the current pilot study, there were 18 participants recruited from a research organization, and a University. Participants did not receive any compensation for their participation. The sample was 55.6% male (10 participants) and 44.4% female (8 participants). Reported participant ages ranged between 18 years and 51 years (*M* = 28.8 years, *SD* = 9.2 years). As there were 3 conditions, there were 6 participants in each condition.

2.2 Design

The current study employed a between subjects design. The independent variable was the types of names included in the tutorial during the learning phase of the study. There were three conditions: Self-Reference, Popular Culture, and Generic. The dependent variables were ratings of self-efficacy before and after the tutorial, ratings of logic grid puzzle experience after the tutorial, performance on a multiple-choice quiz about the content of the tutorial, performance on applied logic puzzle questions (which asked the participants to apply the skill they learned in a new situation), performance on a logic puzzle of the same difficulty as the tutorial, and on one that was more difficult.

2.3 Apparatus

Laptop and GIFT. The study was conducted on a laptop that was on a docking station, and connected to a monitor. GIFT 2.5 and PowerPoint 2007 were installed on the laptop, and a GIFT course was created for each condition of the experiment.

Q-sensor. Participants wore a Q-sensor on their left wrists. It is a small band approximately the size of a watch, which measures electrodermal activity (EDA).

2.4 Procedure

Upon arriving, participants were given an informed consent form, and the opportunity to ask questions. For this pilot study, participation occurred individually. After signing the form, participants were randomly assigned to a condition. The experimenter launched ActiveMQ and the GIFT Monitor on the computer. Participants were then fitted with the Q-sensor on their left wrists. The experimenter clicked “Launch all Modules” and then proceeded to synchronize the Q-sensor with the computer. If synchronization was unsuccessful after three tries, the experimenter edited the GIFT sensor configuration file and changed the sensor to the Self Assessment Monitor as a placeholder (the data from it was not used). Next, the “Launch Tutor Window” button was clicked, and the experiment was launched in Google Chrome. The experimenter created a new UserID for the participant, and then logged in. The correct condition was then selected from the available courses. The participants were then instructed that they should interact with the computer and let the experimenter know if they had any questions.

Participants were first asked to answer a few brief demographics questions (e.g., age/gender) and filled out Compeau and Higgins’ (1995) Self Efficacy Questionnaire (SEQ) with regard to their beliefs in their ability to solve a logic grid puzzle in a computer program and rated their logic grid puzzle experience. They then began the Tutorial. Depending on the condition they were in, they received different instructions in regard to the names to enter (their own name and the name of friends, *Harry Potter* related names, or General names). They then worked through the tutorial that walked them through completing a logic grid puzzle and explained the different types of clues.

After completing the tutorial, they filled in the SEQ again, rated their experience again, and were asked to report any information they remembered from the content of the puzzle. Next, they answered 20 multiple choice questions about the material they learned about in the tutorial. Then, they answered 12 applied clue questions, which provided a story and an individual clue, then asked the participants to select all of the conclusions that could be drawn from that clue. Next, participants had 5 minutes in which to complete an interactive PowerPoint logic grid puzzle at the same level of difficulty as the one that they worked through in the tutorial, and 10 minutes to complete a more difficult puzzle. Finally, they were directed to an external web-site to complete a personality test. They wrote their scores on a piece of paper, and entered them back into GIFT. Afterward, they were given a debriefing form and the study was explained to them.

2.5 GIFT and the Procedure

The Survey Authoring System in GIFT was used to collect survey answers from the participants. While it was a fairly easy to use tool to enter the questions initially, there was some difficulty in the export function. Instead of exporting all the entered questions, there appeared to also be previously deleted questions within the files that were exported. This made it impossible to simply import the questions into an instance of GIFT on an additional computer (in order to have an identical experiment on more than one computer). As a work around, the questions had to be manually typed in and added to each additional computer that was used for the study.

A course file was generated using the Course Authoring Tool. The tool was also fairly easy to use. It provided the ability to author messages that the participant would see between surveys and training applications, determine the specific surveys and PowerPoint applications that would be run, and the order in which they would run. Further, it could send participants to an external web-site; however, while the participants were on the site there was no ability to keep instructions on the screen. Participants only saw a “Continue” button at the bottom of the screen – which may have led to some participants in the current study clicking “Continue” before filling out the surveys they needed to on the web-site. A solution to this was employed by creating a PowerPoint to explain what the participants would be doing on the web-site. However, having the ability to author comments that are seen by the participant while they are on the external web-site would be beneficial.

3 Results

3.1 Pilot Study Results

Performance Results. A series of One Way ANOVAs were run for the percentages correct on the multiple choice questions [$F(2,15) = .389$, $p = .684$], applied clue questions [$F(2,15) = 2.061$, $p = .162$], the easier assessment logic puzzle [$F(2,15) = 3.424$, $p = .060$] and the more difficult logic puzzle [$F(2,15) = 1.080$, $p = .365$]. However, there were no significant differences found between conditions for any of the assessments. See Table 1 for the means and standard deviations for each condition and DV.

	Self-Reference	Popular Culture	Generic
Multiple Choice	*M* = 96.67%, *SD* = 2.58%	*M* = 95.83%, *SD* = 6.65%	*M* = 94.17%, *SD* = 4.92%
Applied Clue	*M* = 80.55%, *SD* = 16.38%	*M* = 87.50%, *SD* = 11.48%	*M* = 69.44%, *SD* = 18.00%
Easy Logic Puzzle	*M* = 51.95%, *SD* = 37.47%	*M* = 93.21%, *SD* = 16.63%	*M* = 74.07%, *SD* = 23.89%
Difficult Logic Puzzle	*M* = 69.78%, *SD* = 24.61%	*M* = 76.89%, *SD* = 16.49%	*M* = 86.89%, *SD* = 19.31%

Table 5. Means and Standard Deviations for Performance Variables for each condition

Logic Grid Puzzle Experience. A 3 (Condition) x 2 (Time of Logic Puzzle Experience) Mixed ANOVA was run comparing the conditions and participant's self rating of their logic grid puzzle experience. Overall, participants indicated that they had significantly higher logic grid puzzle experience after the tutorial (*M* = 3.78, *SD* = 1.215) than before (*M* = 2.00, *SD* = 1.085), *F*(1,15) = 28.764, *p*<.001. However, there was no significant interaction between condition and logic grid puzzle experience ratings, *F*(2, 15) = .365, *p* = .700.

Self Efficacy Questionnaire. A 3 (Condition) x 2 (Time of SEQ score) Mixed ANOVA was run comparing the conditions and the scores on the logic grid puzzle self-efficacy questionnaire. There were significantly higher scores of self-efficacy after tutoring regardless of condition (*M* = 5.583, *SD* = .6564) than before tutoring (*M* = 5.117, *SD* = .7618), *F*(1,15) = 9.037, *p* = .009. However, the condition did not seem to matter, as there was not a significant interaction between condition and time of SEQ score, *F*(2,15) = .661, *p* = .531.

3.2 Using GIFT to extract the information and results

The Event Reporting Tool was used to export survey data from GIFT. However, in the initial GIFT 2.5 version, data from only one participant would export at a time. These files were manually copied and pasted together into one Excel file for analysis. An updated version of GIFT 2.5 offered the ability to export multiple participant files at once. However, if using multiple instances of GIFT on separate computers, it is important to name the questions identically. Combining the outputs of questions that have different names in the survey system may result in the data for those columns not being reported for certain participants.

4 Discussion

4.1 Pilot Results Discussion

The results indicate that the tutorial was successful in teaching participants the skill of completing logic grid puzzles, and made them feel more confident in their abilities than before tutoring. However, the manipulation of the names present in the puzzle during tutoring did not impact performance. As this is a small pilot study, it likely did not have enough power to find results. Currently there are only 6 participants in each condition. The full study is expected to have at least 40 participants in each condition. Individual differences in the ability of individuals to solve the puzzles and the wide variety of ages may also have played a role in the results. Based on the experience with this pilot study, some changes have been made to the full-scale study. First, a pre-test of applied clue questions will be given. Secondly, as not all the participants were able to finish the easier logic puzzle in 5 minutes, the amount of time given for this task will be increased. It is also possible that the current "tests" are not sensitive enough to differences. Further, the sample population for the pilot is different than the intended population for the full-scale study (college students), therefore, those with less research and logic training may show different results.

4.2 GIFT Discussion and Recommendations

GIFT was extremely useful in the current study. During this pilot, participants were able to easily understand and interact with the courses developed with GIFT. All of the survey data was recorded and able to be cleaned up for analysis. One improvement that could be made would be to change the UserID system. Currently, it is set up such that UserIDs are created one by one and in order. It would be beneficial to be able to assign a specific participant number as the User ID in order to reduce confusion when exporting the results (e.g. "P10" rather than "1"). Further, improvements could be made to the ability to launch an external web-site. Currently, there is no ability to provide on-screen directions to individuals while they are on the page. While the Survey Authoring System is useful, it could be greatly improved by having a more reliable import/export option for questions and entire surveys. By doing so, it would be easier to set up identical instances of GIFT on multiple computers.

Overall, GIFT is a useful, cost effective tool which is an asset in running a study. It has a wide variety of helpful functions, and with each release the improvements will likely make it even more valuable to researchers who adopt it.

5 References

1. Anand, P.G., & Ross, S.M. (1987). Using computer-assisted instruction to personalize arithmetic for elementary school children. *Journal of Educational Psychology, 79* (1), 72 – 78.
2. Compeau, D.R., & Higgins, C.A. (1995). Computer self-efficacy: Development of a measure and initial test. *MIS Quarterly, 19* (2), 189 – 211.

3. Chow B. (October 6th, 2010). The quest for deeper learning. *Education Week*, Retrieved from http://www.hewlett.org/newsroom/quest-deeper-learning
4. Ku, H-Y, & Sullivan, H.J. (2002). Student performance and attitudes using personalized mathematics instruction. *ETR&D, 50* (1), 21 – 34.
5. Lombardo, M.V., Barnes, J.L., Wheelwright, S.J., & Baron-Cohen, S. (2007). Self-referential cognition and empathy in autism. *PLOS one, 2* (9), e883.
6. McDonald, K. (2007). Teaching L2 vocabulary through logic puzzles. *Estudios de Linguistica Inglesa Aplicada, 7*, 149 – 163.
7. Moreno, R., & Mayer, R.E. (2000). Engaging students in active learning: The case for personalized multimedia messages: *Journal of Educational Psychology, 92* (4), 724 – 733.
8. Sinatra, A.M., Sims, V.K., Najle, M.B., & Chin, M.G. (2011, September). An examination of the impact of synthetic speech on unattended recall in a dichotic listening task. *Proceedings of the Human Factors and Ergonomics Society, 55*, 1245 – 1249.
9. Sottilare, R.A., Brawner, K.W., Goldberg, B.S., & Holden, H.K. (2012). The Generalized Intelligent Framework for Tutoring (GIFT). Orlando, FL: U.S Army Research Laboratory – Human Research & Engineering Directorate (ARL-HRED).
10. Symons, C.S., & Johnson, B.T. (1997). The self-reference effect in memory: A meta-analysis. *Psychological Bulletin, 121* (3), 371 – 394.

6 Acknowledgment

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-12-2-0019. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

Authors

Anne M. Sinatra: Anne M. Sinatra, Ph.D. is an Oak Ridge Associated Universities Post Doctoral Fellow in the Learning in Intelligent Tutoring Environments (LITE) Lab at the U.S. Army Research Laboratory's (ARL) Simulation and Training Technology Center (STTC) in Orlando, FL. The focus of her research is in cognitive and human factors psychology. She has specific interest in how information relating to the self and about those that one is familiar with can aid in memory, recall, and tutoring. Her dissertation research evaluated the impact of using degraded speech and a familiar story on attention/recall in a dichotic listening task. Prior to becoming a Post Doc, Dr. Sinatra was a Graduate Research Associate with the University of Central Florida's Applied Cognition and Technology (ACAT) Lab, and taught a variety of undergraduate Psychology courses. Dr. Sinatra received her Ph.D. and M.A. in Applied Experimental and Human Factors Psychology, as well as her B.S. in Psychology from the University of Central Florida.

Detection and Transition Analysis of Engagement and Affect in a Simulation-based Combat Medic Training Environment

Jeanine A. DeFalco[1], Ryan S.J.d.Baker[1]

[1]*Teachers College, Columbia University, New York, NY*
jad2234@tc.columbia.edu, baker2@exchange.tc.columbia.edu

Abstract. Developing intelligent tutoring systems that respond effectively to trainee or student affect is a key part of education, particularly in domains where learning to regulate one's emotion is key. Effective affect response relies upon effective affect detection. This paper discusses an upcoming cooperative study between the Army Research Laboratory, Teachers College, Columbia University, and North Carolina State University, with the goal of developing automated detectors that can infer trainee affect as trainees learn by interacting with the vMedic system, which trains learners in combat medicine. In this project, trainee interactions with vMedic will be synchronized with observations of engagement and affect; and physical sensor data on learners, obtained through GIFT's Sensor Module. The result will be models of trainee affect, ready for integration into the GIFT platform, which can leverage sensor information when available, but which can make reasonably accurate inference even without sensor data.

Keywords: GIFT, vMedic, affect, tutoring, intelligent tutoring systems, learning, automated detectors, game-based training

1 Introduction

In recent years, there has been increasing interest in modeling affect within intelligent tutoring systems [7, 11] and using these models to drive affect-sensitive interventions [2]. In this paper, we describe an ongoing collaborative project between the Army Research Laboratory, Teachers College Columbia University, and North Carolina State University, which has the goal of developing automated detection of trainee affect that can leverage sensors when they are available, but which can function robustly even when sensors are not available.

Within this research, trainee affect will be studied in the context of the vMedic, (a.k.a. TC3Sim), a game developed for the U.S. Army by Engineering and Computer Simulations (ECS) in Orlando, Florida, to train combat medics and combat lifesavers on providing care under fire and tactical field care. Trainees will also complete material on hemorrhage control within the auspices of the GIFT framework [12], the Army Research Laboratory's modular framework for Computer-Based Training Systems, with the goal of integrating eventual affect detection into the GIFT framework's User

Module (realized as necessary within the Sensor Module). In turn, the affect detectors will be built into the pedagogies realized through the GIFT Framework's Pedagogical Module, for instance to realize interventions through the embedded instructor and other non-player characters.

In this fashion, this project will contribute not just to the assessment of affect within vMedic, but also to the GIFT framework's broader goal of integrating a range of types of models and detectors into the GIFT framework. By serving as a test case for incorporating two types of detection into GIFT (sensor-free affect detection, and sensor-based affect detection), this project will assist in understanding how GIFT needs to be enhanced to incorporate the full range of models currently being developed by this research community.

Using these detectors, further work will be conducted to study student affective trajectories within vMedic, which affective and engagement states influence learning of the key material within vMedic, and how trainee affect can best be supported based on the results of affect detection. The work to study the relationship between affect, engagement, and outcome variables will provide important evidence on which affective states and engagement variables need to be responded to in a suite of optimally effective computer-based tutoring systems for Army use. Also, integrating automated detectors and interventions into vMedic through GIFT's Trainee Module and Pedagogical Module will provide a valuable example of how to respond to trainees' negative affect and disengagement, a valuable contribution in improving vMedic and similar training systems used by the U.S. Army.

2 Previous Research: Theoretical Grounding

Affect influences learning in at least three ways: memory, attention, and strategy use [16, 18]. Overly strong affect can contribute to cognitive load in working memory, reducing the cognitive resources available to students in learning tasks [13]. Beyond this, negative affective states such as frustration and anxiety can draw cognitive resources away from the task at hand to focus on the source of the emotion [20]. These high-intensity negative affective states can be particularly important for trainees learning content that is emotionally affecting or relevant to their future goals. Combat medicine training for soldiers has each of these components; it is relevant to future situations where they or their fellow soldiers may be in physical danger, and the training in vMedic is designed to be realistic and to involve scenarios where soldiers scream in pain, for example.

However, boredom and disengagement are also relevant to trainees engaging in a task that is not immediately relevant, even if it is relevant to a trainee's longer-term goals. Boredom results in several disengaged behaviors, including off-task behavior [8] and gaming the system [5], when a student intentionally misuses the learning software's help or feedback in order to complete materials without learning. Both gaming the system and off-task behavior have been found to be associated with poorer learning in online learning environments [cf. 4].

However, automated systems that infer and respond to differences in student affect can have a positive impact on students, both in terms of improved affect and improved learning [2, 13]. Similarly, automated interventions based on engagement detection can improve both engagement and learning [2].

A key aspect of automated intervention is the need to detect differences in student affect and engagement, in order to intervene effectively. Recent work has detected these constructs, both using sensors [15], and solely from the student's interactions with the learning system [5, 7, 8]. In recent years, sensor-free models have been developed of a range of behaviors associated with engagement or disengagement: gaming the system [3, 4], off-task behavior [3], self-explanation – an engaged behavior [6], carelessness [18], and WTF ("without thinking fastidiously") behavior, actions within a learning system not targeted towards learning or successful performance [24, 34], among other constructs.

Similarly, automated detectors have been developed that can infer affect solely from student interactions with educational software [7, 10, 11]. However, better performance has typically been achieved by systems that infer affect not only from student interactions, but also from information obtained by physiological sensors. These recognition models use a broad range of physical cues ensuing from affective change. Observable physical cues include body and head posture, facial expressions, and posture, and changes in physiological signals such as heart rate, skin conductivity, temperature, and respiration [1]. In particular, galvanic skin response (GSR) has been correlated with cognitive load and stresses [15], frustration [9], and detecting multiple user emotions in an educational game [10].

3 Project Design

The first step towards developing automated detectors of student affect is to obtain "ground truth" training labels of student affect and engagement. Two approaches are typically chosen to obtain these labels: expert human coding, and self-report [11]. In this project, we rely upon expert human coding, as self-report can be intrusive to the processes we want to study, and self-presentation and demand effects are also likely to be of concern with the population being studied (military cadets are unlikely to want to report that they are frustrated or anxious).

These training labels will be collected in a study to be conducted at West Point, the United States Military Academy. Each trainee will use vMedic for one hour in a computer laboratory, in groups of ten at a time. The following sources of student data will be collected: field observations of trainee affect and engagement; the Immersive Tendencies Questionnaire (ITQ), an instrument to gauge an individual's propensity to experience presence in mediated environments a priori to system interaction; the Sense of Presence questionnaire, a 44-item questionnaire that rates subjective levels of presence on four separate factors: (1) Sense of Physical Space (19 items); (2) Engagement (13 items); (3) Ecological Validity/Naturalness (5 items); and (4) Negative Effects (6 items) [19];, a pre-and post test on hemorrhage control (a total of 12 questions, same questions used in pre-and post-test, though ordered differently), and physical sensor data for students as they play the game. The following physical sensors will be used: Q-sensors, and Kinect depth sensors. Q-sensors track skin conductance data, a measure of arousal, while Kinect depth sensors record depth-map images to support recognition of postural positions.

Within this study, expert codes of trainee affect and engagement will be collected by a trained coder (the first author) using the BROMP 1.0 field observation protocol [16]. The field observations will be conducted in a pre-chosen order to balance observation across trainees and avoid bias towards more noteworthy behaviors or affect. Observations will be conducted using quick side glances in order to make it less clear when a specific trainee is being observed. Coding includes recording the first behavior and affect displayed by the trainee within 20 seconds of the observation, choosing from a predetermined coding scheme. The affect to be coded includes: frustration, confusion, engaged concentration [5], boredom, anxiety, and surprise. Affect will be coded according to a holistic coding scheme. Behavior coding includes: on-task behavior, off-task behavior, gaming the system, "psychopath" behavior (friendly fire, killing bystanders), and WTF ("without thinking fastidiously") behavior, where the trainee's actions have no relation to the scenario [17]. In order to be BROMP-certified, a coder must achieve inter-rater reliability of 0.6 or higher to another BROMP-certified coder; two coders are currently trained at Teachers College, and are available for the project.

Field observation coding will be conducted within a handheld Android app, HART, designed for this purpose [7]. The field observations will be synchronized to the other data sources, based on use of an internet time server. Synchronization will be with reference to several data sources, including trainee interactions with vMedic, provided through the GIFT framework's Trainee Module, and physical sensor data on learners, obtained through GIFT's Sensor Module. We anticipate synchronization to involve a skew of 1-2 seconds, based on the time required to enter observations. The GIFT platform includes a synchronization library, which connects to an Internet time-server so that a precise time-stamp can be added to the logs of trainee interactions with vMedic, and the corresponding sensor data. By connecting to the exact same timeserver, the interactions with vMedic, field observations of engagement and affect, and physical sensor data on learners, three data sources can be precisely synchronized.

Automated detectors will be developed using the interaction logs alone, for use when physiological sensors are not available, and using the sensors as well, for situations where they are. A standard approach of conducting feature engineering and then developing classifiers, and validating the classifiers using student-level cross-validation, will be used.

4 Conclusion

The current project has the goal of enhancing the GIFT framework through the creation of models that can infer trainee engagement and affect. This project is expected to both enhance the capacities of the vMedic software, and to provide a model for how this type of detection can be integrated into the GIFT framework more generally. As such, this project is just one small component of the larger efforts that are currently being pursued by the Army Research Lab, to make the GIFT framework a general and extensible platform to achieve the US Army's overall objective of applying learning theory and state-of-the-art learning technology to achieve superior training results for warfighters [14]. We anticipate that this collaborative effort will provide useful information on the future enhancement of the GIFT platform; as such, this project represents a step towards the vision of adaptable and scalable Computer-Based Training Systems helping to enhance the training of U.S. Army military personnel and prepare U.S. soldiers for the conflicts of the future.

Acknowledgments. We thank our research colleagues at the Army Research Lab, Dr. Robert Sottilare, Benjamin Goldberg and Keith Brawner, as well as at North Carolina State University, Dr. James Lester, Dr. Bradford Mott, and Jonathan Rowe. This research is supported by a grant by the Army Research Lab. Any opinions, findings, and conclusions expressed in this paper are those of the authors and do not necessarily reflect the views of ARL or NCSU.

5 References

1. Allanson, J., Fairclough, S. H. A research agenda for physiological computing. In Interacting with Computers, 16 (2004), 857–878.
2. Arroyo, I., Ferguson, K., Johns, J., Dragon, T., Meheranian, H., Fisher, D., Barto, A., Mahadevan, S.,Woolf. B. Preparing disengagement with non-invasive interventions. In Proceedings of the 13th International Conference on Artificial Intelligence in Education,.(2007) 195–202.
3. Baker, R.S.J.d. Is Gaming the System State-or-Trait? Educational Data Mining Through the Multi-Contextual Application of a Validated Behavioral Model. In Proceedings of the Workshop on Data Mining for User Modeling at the 11th International Conference on User Modeling 2007, 76-80.
4. Baker, R.S., Corbett, A.T., Koedinger, K.R., Wagner, A.Z. Off-Task Behavior in the Cognitive Tutor Classroom: When Students "Game The System." In Proceedings of ACM CHI 2004: Computer-Human Interaction 2004, 383-390.
5. Baker, R.S.J.d., D'Mello, S.K., Rodrigo, .M.T., Graesser, A.C. Better to Be Frustrated than Bored: The Incidence, Persistence, and Impact of Learners' Cognitive-Affective States during Interactions with Three Different Computer-Based Learning Environments. In International Journal of Human-Computer Studies, 48 (2010), 223-241.
6. Baker, R.S.J.d., Gowda, S.M., Corbett, A.T. Automatically Detecting a Student's Preparation for Future Learning: Help Use is Key. In Proceedings of the 4th International Conference on Educational Data Mining 2011, 9-188.
7. Baker, R.S.J.d., Gowda, S.M., Wixon, M., Kalka, J., Wagner, A.Z., Salvi, A., Aleven, V., Kusbit, G., Ocumpaugh, J., Rossi, L. Sensor-free automated detection of affect in a Cognitive Tutor for Algebra. In Proceedings of the 5th International Conference on Educational Data Mining 2012, 126-133.
8. Baker, R.S.J.d., Moore, G., Wagner, A., Kalka, J., Karabinos, M., Ashe, C., Yaron, D. The Dynamics Between Student Affect and Behavior Occuring Outside of Educational Software. In Proceedings of the 4th bi-annual International Conference on Affective Computing and Intelligent Interaction 2011.
9. Burleson, W. Affective learning companions: strategies for empathic agents with real-time multimodal affect sensing to foster meta-cognitive and meta-affective approaches to learning, motivation, and perseverance. In unpublished Doctoral Dissertation. Cambridge, MA: Massachusetts Institute of Technology (2006).
10. Conati, C. Probabilistic Assessment of User's Emotions in Educational Games. In Journal of Applied Artificial Intelligence, 16 (2002), 555-575.

11. D'Mello, S.K., Craig, S.D., Witherspoon, A. W., McDaniel, B. T., Graesser, A. C. Automatic Detection of Learner's Affect from Conversational Cues. In User Modeling and User Adapted Interaction, 18 (2008), 45-80.
12. Goldberg, B., Brawner, K., Sottilare, R, Tarr, R., Billings, D., & Malone, N. Use of Evidence-based Strategies to Enhance the Extensibility of Adaptive Tutoring Technologies. In Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2012.
13. Linnenbrink, E.A., Pintrich, P.R. Multiple Pathways to Learning and Achievement: The Role of Goal Orientation in Fostering Adaptive Motivation, Affect, and Cognition. In Intrinsic and Extrinsic Motivation: The Search for Optimal Motivation and Performance, C. Sansone & J.M. Harackiewicz, Eds. Academic Press, San Diego, 2000, 195-227.
14. McQuiggan, S., Rowe, J., Lee, S., Lester, J. Story-based Learning: The Impact of Narrative on Learning Experiences and Outcomes. In Proceedings of the Ninth International Conference on Intelligent Tutoring Systems (2008), 530-539.
15. Mohammad, Y., & Nishida, T. Using physiological signals to detect natural interactive behavior. In Spring Science & Business Media, 33(2010) 79-92.
16. Ocumpaugh, J., Baker, R.S.J.d., Rodrigo, M.M.T. Baker-Rodrigo Observation Method Protocol (BROMP) 1.0. Training Manual version 1.0. Technical Report. EdLab: New York, NY, Manila, Philippines: Ateneo Laboratory for the Learning Sciences (2012).
17. Sabourin, J., Rowe, J., Mott, B., and Lester, J. When Off-Task in On-Task: The Affective Role of Off-Task Behavior in Narrative-Centered Learning Environments. Proceedings of the 15th International Conference on Artificial Intelligence in Education, pp. 534-536, 2011.
18. San Pedro, M.O.C., Rodrigo, M.M., Baker, R.S.J.d. The Relationship between Carelessness and Affect in a Cognitive Tutor. In Proceedings of the 4th bi-annual International Conference on Affective Computing and Intelligent Interaction 2011.
19. Witmer, B.G. & Singer, M.J. Measuring presence in virtual environments: A presence questionnaire. In Presence, 7 (1998), 225-240.
20. Zeidner, M. Test Anxiety: The State of the Art. Springer, Berlin, 1998.

Authors

Jeanine A. DeFalco is a Doctoral Research Fellow in Cognitive Studies at Teachers College, Columbia University. Jeanine's research interests include embodied cognition and role-play as a methodology for improving analogic reasoning and creative problem solving in both live and simulated learning platforms. A member of Kappa Delta Pi, the international honors society for education, Jeanine has a Masters in Educational Theatre, Colleges and Communities, from the Steinhardt School, New York University, and a Masters in Drama Studies from The Johns Hopkins University. Jeanine's paper, "Cognition, O'Neill, and the Common Core Standards," has an expected publication in the *Eugene O'Neill Journal* for Fall 2013, and she will be presenting this same paper at the July 2013 American Alliance for Theatre and Education conference in Bethesda, MD. Other conference presentations include "Drama as an epistemology for pre-service teachers" forW the 2012 National Communication Association conference in Orlando, FL, and "Teaching O'Neill" at the 8th International Eugene O'Neill Conference, 2011, in New York, NY.

Ryan SJ.d. Baker is the Julius and Rosa Sachs Distinguished Lecturer at Teachers College, Columbia University. He earned his Ph.D. in Human-Computer Interaction from Carnegie Mellon University. Baker was previously Assistant Professor of Psychology and the Learning Sciences at Worcester Polytechnic Institute, and he served as the first Technical Director of the Pittsburgh Science of Learning Center DataShop, the largest public repository for data on the interaction between learners and educational software. He is currently serving as the founding President of the International Educational Data Mining Society, and as Associate Editor of the Journal of Educational Data Mining. His research combines educational data mining and quantitative field observation methods in order to better understand how students respond to educational software, and how these responses impact their learning. He studies these issues within intelligent tutors, simulations, multi-user virtual environments, and educational games.

Run-Time Affect Modeling in a Serious Game with the Generalized Intelligent Framework for Tutoring

Jonathan P. Rowe, Eleni V. Lobene, Jennifer L. Sabourin,
Bradford W. Mott, and James C. Lester

Department of Computer Science, North Carolina State University, Raleigh, NC 27695
{jprowe, eleni.lobene, jlrobiso, bwmott, lester}@ncsu.edu

Abstract. Affective computing holds significant promise for fostering engaging educational interactions that produce significant learning gains. Serious games are particularly well suited to promoting engagement and creating authentic contexts for learning scenarios. This paper describes an ongoing collaborative project between the Army Research Lab (ARL), Teachers College Columbia University, and North Carolina State University to investigate generalized run-time affect detection models in a serious game for tactical combat casualty care, vMedic. These models are being developed and integrated with ARL's Generalized Intelligent Framework for Tutoring (GIFT). Drawing upon our experience with GIFT, we outline opportunities for enhancing GIFT's support for developing and studying run-time affect modeling, including extensions that enhance affective survey administration, leverage mathematical models for formative assessment, and streamline affect data processing and analysis.

Keywords: Affect Detection, GIFT, Serious Games.

1 Introduction

The past decade has witnessed major advances in research on computational models of affect, endowing software systems with affect-sensitivity and yielding new insights into artificial and human intelligence [1]. Education and training have served as key application areas for computational models of affect, producing intelligent tutoring systems (ITSs) that can model students' affective states [2], model virtual agents' affective states [3], and detect student motivation and engagement [4]. Education-focused work on affective computing has sought to increase the fidelity with which affective and motivational processes are understood and utilized in ITSs in an effort to increase the effectiveness of tutorial interactions and, ultimately, learning.

The rise of affective computing has coincided with growing interest in digital games for learning. Serious games have emerged as an effective vehicle for learning and training experiences [5]. The education community has developed a broad range of serious games that combine pedagogy and interactive problem solving with the salient properties of games (e.g., feedback, challenge, rewards) to foster motivation and engagement [6–8]. Efforts to design serious games for training have also been the subject of increasing interest in the defense community [6, 9].

A notable property of serious games is their potential to serve as virtual laboratories for studying affect in learning and training applications. Serious games are well suited to promoting high levels of learner engagement and providing immersive training experiences. These features can have significant impacts on learners' affective trajectories, as well as the relationships between learners' affect and performance. For example, in training tasks that evoke considerable stress or anxiety it is plausible that serious games may foster affective experiences that differ considerably from non-mission-critical domains, significantly impacting learners' abilities to successfully demonstrate their knowledge. Salient features such as these raise questions about how to most effectively study and model learner affect during interactions with serious games, as well as questions about how these methods and models can be generalized to other training environments and domains.

In this paper we describe a collaborative project with Teacher's College Columbia University (TC) and the Army Research Lab (ARL) that uses the Generalized Intelligent Framework for Tutoring (GIFT) to investigate run-time affect modeling in a serious game for tactical combat casualty care. The project draws on recent advances in five areas: minimally-obtrusive and synchronize-able field observations of learner affect [10], empirical studies of serious games [7], educational data mining of affect logs [11–12], hardware sensor-based measurements of affect [13], and generalized intelligent tutoring frameworks [14]. The project's objectives are two fold: 1) create modular intelligent tutor components for run-time affect modeling that generalize across multiple training environments and scale to alternate hardware configurations, and 2) develop tools and procedures to facilitate future research on affective computing in learning technologies. This paper focuses on North Carolina State University's component of the project, which emphasizes sensor-based affect detection, and it outlines recommendations for future enhancements to GIFT in support of run-time affect modeling. Specifically, we outline several opportunities for extending GIFT, which include incorporating support for temporal models of affect such as affect transitions; expanding GIFT's survey tools to serve as a centralized repository of validated instruments with an integrated web-based infrastructure for administering surveys; taking advantage of item response theory techniques to conduct stealth, formative assessment of trainee attitudes during learning interactions; and incorporating features to streamline affect data post-processing.

2 Investigating Affect in a Serious Game for Tactical Combat Casualty Care

The goal of our collaboration with ARL and TC is to model trainee affect in a serious game for combat medic training, vMedic, using GIFT. The research team will utilize machine-learning techniques to induce models for detecting trainee's affective states and levels of engagement during interactions with the vMedic software. Affect and engagement significantly influence learning, and we hypothesize that this will be especially true for the vMedic training environment due to the time-sensitive, life-or-

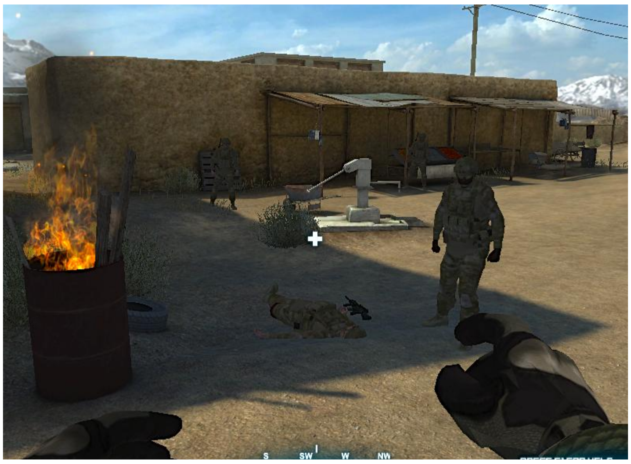


Fig. 9. vMedic serious game for tactical combat casualty care.

death decisions inherent in tactical combat casualty care. In combination with field observations of trainee affect and trace data from the vMedic serious game, the North Carolina State University team will investigate data streams produced by a Microsoft Kinect sensor and Affectiva Q-Sensor to develop and validate affect detection models. The research team seeks to produce models that 1) integrate trace data logs, sensor data, and field observations of trainee emotions; 2) predict emotions accurately and efficiently when hardware sensors are available; and 3) scale gracefully to settings where hardware sensors are unavailable. The models will be developed and utilized to improve trainee engagement and affect when using vMedic, and they will be integrated with interaction-based models devised by colleagues at TC.

The curriculum for the study focuses on a subset of skills for tactical combat casualty care: care under fire, hemorrhage control, and tactical field care. The study materials, including pre-tests, training exercises, and post-tests, are managed entirely by GIFT, which supports inter-module communication through its service-oriented architecture. At the onset of training, learners are presented with direct instruction about tactical combat casualty care in the form of a PowerPoint presentation. After completing the PowerPoint, participants play through a series of scenarios in the vMedic serious game. vMedic presents combat medic scenarios from a first-person perspective (Fig. 1). The learner adopts the role of a combat medic faced with a situation where one (or several) of his fellow soldiers has been seriously injured. The learner is responsible for properly treating and evacuating the casualty. The scenarios include the following elements: a tutorial level for trainees to learn the controls and game mechanics of vMedic; a scenario focusing on a lower leg amputation; a vignette about a patrol that leads to several casualties; and the “Kobayashi Maru” scenario where the trainee cannot save the casualty’s life regardless of her course of medical treatment. vMedic is currently being used at scale by the U.S. Army for combat medic training, and it has been integrated with GIFT by ARL.

The focus of North Carolina State University’s part of the project is leveraging hardware sensor data from a Microsoft Kinect for Windows and Affectiva Q-Sensor to generate affect detection models. Both hardware sensors are integrated with GIFT, enabling the sensor data to be automatically synchronized with vMedic and PowerPoint interaction logs. This architecture removes the need to directly integrate hardware sensors with individual learning environments. Whenever a new training environment is integrated with GIFT, no additional work is required to use the hardware sensors with the new environment.

The Microsoft Kinect provides four data channels: skeleton tracking, face tracking, RGB (i.e., color), and depth. The first two channels leverage built-in tracking algorithms (which are included with the Microsoft Kinect for Windows SDK) for recognizing a user's skeleton, represented as a graph with vertices as joints, and a user's face, represented as a three-dimensional polygonal mesh. The skeleton and face models can move, rotate, and deform based on the user's head movements and facial expressions. The RGB channel is a 640x480 color image stream comparable to a standard web camera. The depth channel is a 640x480 IR-based image stream depicting distances between objects and the sensor. The latter two channels produce large quantities of uncompressed image data, so configuration options have been added to GIFT to adjust the sample rate (default is 30 Hz), sample resolution, and compression technique. RGB and depth data can be stored in an uncompressed format, in PNG format with zlib compression, or in PNG format with lz4 compression. We intend to utilize data from the Microsoft Kinect to detect user posture, hand gestures, and facial expression. The Affectiva Q-Sensor is a wearable arm bracelet that measures participants' electrodermal activity (i.e., skin conductance), skin temperature, and its orientation through a built-in 3-axis accelerometer. The wireless sensor collects data at 32Hz, and will primarily be used for real-time arousal detection.

Since all technology components in the planned study are managed by GIFT, we have leveraged GIFT's built-in authoring tools to specify the study questionnaires and curriculum tests for assessing trainee knowledge and engagement before and after the learning intervention. We have utilized GIFT's Survey Authoring Tool to rapidly integrate standard presence and intrinsic motivation questionnaires. Additionally, we have used GIFT's sizable repository of reusable content assessment items to create a curriculum test for measuring learning gains across the training sequence.

After specifying the required measures, we used GIFT's Course Authoring Tool to encode the sequence of training and assessment materials that will be presented by GIFT. The Course Authoring Tool includes support for authoring web-based messages that provide instructions to participants, specifying the presentation order of pre- and post-intervention questionnaires and content tests, and specifying the sequence of PowerPoint and vMedic learning activities that occur during the study. When participants take the course, each of these steps is automatically triggered, monitored, and logged by GIFT. It should be noted that authored courses and questionnaires can be easily exported and shared between groups, consistent with GIFT's objective of fostering reusable components.

Currently, our team has established the initial data collection's study procedure, we have tested the integrated hardware sensors, and ensured the reliability of the study's technology setup. In addition to pilot testing field observation tools from TC with GIFT, we are in the process of planning a study at the U.S. Military Academy to investigate cadets' affective experiences during interactions with vMedic.

3 Extending GIFT's Capabilities for Run-Time Affect Modeling

GIFT consists of a suite of software tools, standards, and resources for creating, deploying, and studying modular intelligent tutoring systems with re-usable components. GIFT provides three primary functions: authoring capabilities for constructing learning technologies, instruction that integrates tutorial principles and strategies, and support for evaluating educational tools and frameworks. These capabilities provide the foundation for our investigation of generalizable run-time affect models. This section discusses several areas for which extensions to GIFT could support research and development of generalized affect models in ITSs.

3.1 Detecting and Understanding Learner Affect

While considerable work remains to identify the precise cognitive and affective mechanisms underlying learning, significant progress has been made in identifying the emotions that students commonly experience and how these affect the learning process. For instance, both D'Mello *et al*. [15] and Baker *et al*. [16] have shown that students are most likely to remain in the same emotion state over time and that certain emotional transitions are more likely than others. Students who are experiencing *boredom* are much more likely to experience *frustration* immediately following the state of *boredom* than they are to enter into positive learning states such as *flow* [15–16]. In this way affect transition analyses reveal underlying relationships between affect and learning, which occur generally across intelligent tutoring systems.

Since existing research has suggested that affect transition analysis is both a useful and generalizable tool for investigating learner emotions, incorporating affect transition models within GIFT is a natural direction for future research and development. This will likely raise questions about how to effectively, and generally, integrate affect transition modeling capabilities with each tutor module in the GIFT architecture: the sensor module, learner module, pedagogical module, and domain module. When designing these components, one must consider how these components communicate with one another, and how the system should be configured to support cases where affect-sensitive components are missing. For example, physiological sensors are highly beneficial for affect recognition, but may not be available in all cases. Consequently, a learner model relying on output from such a sensor would need to be adapted, or gracefully deactivated, in a manner that minimizes negative impacts on other modules. Similarly, different genres of serious games have distinct capabilities and affordances. For example, serious games with believable virtual agents may present different opportunities for affective feedback than serious games without virtual agents. Pedagogical modules should possess mechanisms for handling cases where alternate learning environments support different types of interventions.

3.2 Advances in Survey Administration using GIFT

In the educational research community there is a persistent need for streamlined instrument access, validation, and administration. GIFT currently provides a rich collection of content test items and questionnaires that can be re-used across studies and training environments. This survey repository could be expanded to serve the broader research community by systematically adding validated assessment measures and questionnaires used by the education community. GIFT could serve as a searchable, centralized repository of validated instruments, with an integrated web-based infrastructure for administering surveys before and after learning interventions. The instruments could be submitted and listed by category with their important item information such as validity and reliability, links to published papers that describe how the instruments have been used in prior studies, and specific instructions regarding their appropriate use. This type of integrated resource for obtaining, evaluating, and administering questionnaires, content tests, and surveys could help streamline the community's affective computing research efforts, and serve as an entry point for researchers to begin using GIFT. Researchers commonly spend significant effort trying to locate information about study instruments, and GIFT could serve as a tool to facilitate the survey development and selection process. While other domain specific instrument collections are freely available (e.g., www.IPIPori.org [17]), they do not include features for integrating instruments into surveys, or administering surveys to users. GIFT could also reduce the time allocated to integrating surveys with systems such as SurveyMonkey or Qualtrics while encouraging the use of high quality validated instruments.

3.3 Leveraging Mathematical Models for Formative Assessment in GIFT

Building on the availability of this instrument database, there are opportunities to take advantage of item response theory techniques to conduct stealth, formative assessment during learning interactions. Item response theory (IRT) is a mathematical framework for performing measurement in which the variable is continuous in nature while allowing for an individual person and item to be mapped on the same latent trait continuum [18]. An ideal point response process is an IRT approach based on the idea that an individual only endorses an item if he or she is located close to the item on a latent continuum [19]. In other words, if an item is too extreme in either direction, the individual will respond negatively to the item. It can be used with both dichotomous (e.g., content knowledge test) and polytomous data (e.g., Likert-type attitudes or personality [19]). GIFT is well positioned to integrate ideal point methods within user experiences for stealth and ongoing assessment. To date, little research has investigated embedding adaptive, formative assessment within serious games using intermittent item presentation through ideal point methods with a rich database of instruments from which to select.

GIFT offers the opportunity for assessment of both knowledge and attitudinal (e.g., affective states) variables within immersive training experiences. Using GIFT's capabilities, single items can be "transmitted" as part of the story line within a game experience to the participant. GIFT can run mathematical models in the background to determine the best item to present at the next natural point. Conceptually this approach is similar to computerized adaptive tests designed by major test development companies. For example, if the participant responds negatively to the question "I want to repeat this activity over and over," he or she can be presented with an item lower on the latent trait continuum (e.g., "This activity is interesting for now"). GIFT, having access to all of the item information for each potential question, can strategically present a series of them. By the end of the serious game experience, rich data regarding the individuals' location on a latent trait continuum (e.g., engagement) would be available.

3.4 Streamlined Data Processing and Analysis with GIFT

Another opportunity for GIFT to address practical challenges in affective computing research is in post-processing data. Better solutions are needed for merging files and cases and quickly ascertaining basic information from data sets. GIFT could potentially mitigate some of these challenges by introducing standards for data collected during different stages of research; typically data from different stages is encoded in a variety of formats, and a considerable amount of labor is dedicated to data integration after a study has been completed. GIFT could provide a service that automatically links pre, during, and post data for individual participants, thereby reducing labor in data cleaning and transformation steps. GIFT could also be extended to offer quick summary statistics and perform simple operations such as summarizing demographics, computing composite scores for instruments, and providing general summary results. These tools would be especially helpful with affective instruments that often require reverse scoring and other manipulations prior to analysis.

4 Conclusion

This paper has described a collaborative project between the Army Research Lab, Teachers College Columbia University, and North Carolina State University that aims to investigate run-time affect modeling in a serious game for combat medic training, vMedic. In addition to describing this project, we have outlined a number of ways to extend GIFT's capabilities to improve affective computing research for educational applications. We anticipate that these opportunities could increase GIFT's future impact and usage as a tool for ITS researchers.

Acknowledgments. The authors wish to thank the Army Research Lab for supporting this research. Additionally, we wish to thank our collaborators, Ryan Baker and Jeanine DeFalco at Teacher's College Columbia University, and Bob Sottilare, Keith Brawner, Benjamin Goldberg, and Heather Holden from ARL.

5 References

1. Calvo, R. A. & D'Mello, S.K.: Affect Detection: An Interdisciplinary Review of Models, Methods, and their Applications. In: IEEE Transactions on Affective Computing, pp. 18–37. (2010)
2. Conati, C., Maclaren, H.: Empirically Building and Evaluating a Probabilistic Model of User Affect. User Modeling and User-Adapted Interaction. 19, 267–303 (2009)
3. Marsella, S.C., Gratch, J.: EMA: A Process Model of Appraisal Dynamics. Cognitive Systems Research. 10, 70–90 (2009)

4. Forbes-Riley, K., Litman, D.: When Does Disengagement Correlate with Performance in Spoken Dialog Computer Tutoring? International Journal of Artificial Intelligence in Education. (2013)
5. Wouters, P., Van Nimwegen, C., Van Oostendorp, H., & Van der Spek, E.D.: A Meta-Analysis of the Cognitive and Motivational Effects of Serious Games. Journal of Educational Psychology. (2013)
6. Johnson, W.L.: Serious Use of a Serious Game for Language Learning. International Journal of Artificial Intelligence in Education. 20, 175–195 (2010)
7. Rowe, J P, Shores, L. R., Mott, B. W., & Lester, J.C.: Integrating Learning, Problem Solving, and Engagement in Narrative-Centered Learning Environments. International Journal of Artificial Intelligence in Education. 21, 166–177 (2011)
8. Halpern, D., Millis, K., Graesser, A.: Operation ARA: A computerized learning game that teaches critical thinking and scientific reasoning. Thinking Skills and Creativity. 7, 93–100 (2012)
9. Kim, J., Hill, R. W., Durlach, P. J., Lane, H. C., Forbell, E., Core, M., Marsella, S.C., et al.: BiLAT: A game-based environment for practicing negotiation in a cultural context. International Journal of Artificial Intelligence in Education. 19, 289–308 (2009)
10. Rodrigo, M.M.T., Baker, R.S.J.d., Agapito, J., Nabos, J., Repalam, M.C., Reyes, S.S., San Pedro, M.C.Z.: The Effects of an Interactive Software Agent on Student Affective Dynamics while Using an Intelligent Tutoring System. In: IEEE Transactions on Affective Computing, pp. 224–236. (2012)
11. Pardos, Z.A., Baker, R.S.J.d., San Pedro, M.O.C.Z., Gowda, S.M., Gowda, S.M.: Affective states and state tests: Investigating how affect throughout the school year predicts end of year learning outcomes. In: Proceedings of the 3rd International Conference on Learning Analytics and Knowledge, pp. 117–124. (2013)
12. Sabourin, J., Mott, B.W., Lester, J.C.: Modeling Learner Affect with Theoretically Grounded Dynamic Bayesian Networks. In: Proceedings of the 4th International Conference on Affective Computing and Intelligent Interaction, pp. 286–295. (2011)
13. Grafsgaard, J.F., Boyer, K.E., Lester, J.C.: Predicting Facial Indicators of Confusion with Hidden Markov Models. In: Proceedings of the 4th International Conference on Affective Computing and Intelligent Interaction, pp. 97–106. (2011)
14. Sottilare, R. A., Goldberg, B. S., Brawner, K. W., & Holden, H.K.: A modular framework to support the authoring and assessment of adaptive computer-based tutoring systems (CBTS). In: Proceedings of the Interservice/Industry Training, Simulation, and Education Conference, (2012)
15. D'Mello, S., Taylor, R.S., Graesser, A.C.: Monitoring Affective Trajectories during Complex Learning. In: Proceedings of the 29th Annual Meeting of the Cognitive Science Society, pp. 203–208. (2007)
16. Baker, R., Rodrigo, M., Xolocotzin, U.: The dynamics of affective transitions in simulation problem-solving environments. In: Proceedings the 2nd International Conference on Affective Computing and Intelligent Interactions, pp. 666–677. (2007)
17. Goldberg, L. R., Johnson, J. A., Eber, H. W., Hogan, R., Ashton, M. C., Cloninger, C. R., & Gough, H.C.: The International Personality Item Pool and the future of public-domain personality measures. Journal of Research in Personality. 40, 84–96 (2006)
18. de Ayala, R.J.: The Theory and Practice of Item Response Theory. Guilford Press, New York, NY (2009).
19. Stark, S., Chernyshenko, O.S., Drasgow, F., Williams, B.A.: Examining Assumptions About Item Responding in Personality Assessment: Should Ideal Point Methods Be Considered for Scale Development and Scoring? Journal of Applied Psychology. 91, 25–39 (2006)

Authors

Jonathan P. Rowe: Jonathan Rowe is a Research Scientist in the Department of Computer Science at North Carolina State University. He received his Ph.D. in Computer Science from North Carolina State University in 2013. His research is in the areas of artificial intelligence and human-computer interaction for advanced learning technologies, with an emphasis on game-based learning environments. He is particularly interested in intelligent tutoring systems, user modeling, educational data mining, and computational models of interactive narrative. Jonathan has led development efforts on several game-based learning projects, including Crystal Island: Lost Investigation, which was nominated for Best Serious Game at the 2012 Unity Awards and the 2012 I/ITSEC Serious Games Showcase and Challenge. His research has also been recognized with several best paper awards, including best paper at the Seventh International Artificial Intelligence and Interactive Digital Entertainment Conference and best paper at the Second International Conference on Intelligent Technologies for Interactive Entertainment.

Eleni V. Lobene: Eleni Lobene is a Research Psychologist in the Center for Educational Informatics at North Carolina State University. She received her Ph.D. in Industrial/Organizational Psychology from North Carolina State University in 2011, where her research focused on K-12 teacher motivations and perceptions. With a background in psychometrics, including classical test theory and item response theory, Dr. Lobene is interested in study design, instrument validation, assessment, and K-16 education. Prior to joining the IntelliMedia group, she worked as a research assistant at the Friday Institute for Educational Innovation, assisting in the assessment of game-based learning environment effectiveness. She has also served as a primary instructor for undergraduate courses in the Department of Psychology at North Carolina State University and provided consulting services to local and global organizations focusing on improving organizational efficiency and implementing data-based change.

Jennifer L. Sabourin: Jennifer Sabourin is currently a Ph.D. student at North Carolina State University in the Department of Computer Science. She received her B.S. (2008) and M.S. (2012) in Computer Science from North Carolina State University where she graduated as Valedictorian. She is a recipient of the National Science Foundation Graduate Research Fellowship award. Since 2007 she has been engaged in research examining affective and metacognitive issues associated with intelligent learning technologies. Her research efforts have resulted in over 30 published journal articles, book chapters, and refereed conference proceedings. Her work has been recognized with a Best Student Paper Award at the International Conference on Affective Computing and Intelligent Interaction and several additional nominations. In addition to her research on intelligent technologies for education, Sabourin has played an active role in efforts to broaden participation in computing and STEM fields through informal learning activities. She developed and led a year-long middle school program for introducing young students to computing principles. This program has served over 250 middle school students in the last six years and has been successfully disseminated to other schools and districts.

Bradford W. Mott: Bradford Mott is a Senior Research Scientist at the Center for Educational Informatics in the College of Engineering at North Carolina State University. He received his Ph.D. in Computer Science from North Carolina State University in 2006, where his research focused on computational models of interactive narrative. His research interests include intelligent game-based learning environments, computer games, and intelligent tutoring systems. Prior to joining North Carolina State University, he worked in the game industry developing cross-platform middleware solutions for the PlayStation 3, Wii, and Xbox 360. In 2000, he co-founded and was VP of Technology at LiveWire Logic where he led development efforts on the RealDialog™ product suite, an automated natural language customer service solution leveraging corpus-based computational linguistics.

James C. Lester: James Lester is Distinguished Professor of Computer Science at North Carolina State University. His research focuses on transforming education with technology-rich learning environments. Utilizing AI, game technologies, and computational linguistics, he designs, develops, fields, and evaluates next-generation learning technologies for K-12 science, literacy, and computer science education. His work on personalized learning ranges from game-based learning environments and intelligent tutoring systems to affective computing, computational models of narrative, and natural language tutorial dialogue. He has served as Program Chair for the International Conference on Intelligent Tutoring Systems, the International Conference on Intelligent User Interfaces, and the International Conference on Foundations of Digital Games, and as Editor-in-Chief of the International Journal of Artificial Intelligence in Education. He has been recognized with a National Science Foundation CAREER Award and several Best Paper Awards.

Toward a Generalized Framework for Intelligent Teaching and Learning Systems: The Argument for a Lightweight Multiagent Architecture

Benjamin D. Nye and Donald M. Morrison

Institute for Intelligent Systems, The University of Memphis, Memphis, Tennessee
{bdnye and dmmrrson}@memphis.edu

Abstract The U.S. Army's Generalized Intelligent Framework for Tutoring (GIFT) is an important step on the path toward a loosely coupled, service-oriented system that would promote shareable modules and could underpin multiagent architectures. However, the current version of the system may be "heavier" than it needs to be and not ideal for new or veteran ITS developers. We begin our critique with a discussion of general principles of multiagent architecture and provide a simple example. We then look at the needs of ITS developers and consider features of a general-purpose framework which would encourage message-driven, multiagent designs, sharing of services, and porting of modules across systems. Next, we discuss features of the GIFT framework that we believe might encourage or discourage adoption by the growing ITS community. We end by offering three recommendations for improvement.

1 Introduction

As the term is used in a seminal paper on the subject, "Is it an agent, or just a program?" (Franklin & Graesser, 1997), an autonomous agent is

> *..a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to affect what it senses in the future. (p. 25)*

Because a human is also an agent according to this definition, in a sense any intelligent tutoring system may be considered a multiagent system (MAS), designed to support interactions between two agents—the user and the intelligent tutor. However, recent years have seen an increasing interest in the development of systems with multiagent architectures in the more interesting sense that functionality is decentralized across different software agents. In this paradigm, each agent has its own knowledge base (set of beliefs), and carries out different tasks, either autonomously or at the request of other agents. Agent-oriented services build on component-based approaches by giving each component distinct goals that it works to fulfill. As a result, the intelligent behavior of the system as a whole emerges from the collective behavior of the individual agents—including, of course, the human user—allowing for what has been called “autonomous cooperation” (Hülsmann, Scholz-Reiter, Freitag, Wucisk, & De Beer, 2006; Windt, Böse, & Philipp, 2005). For recent examples of ITSs that employ multiagent architectures, see Bittencourt et al., 2007; Chen & Mizoguchi, 2004; El Mokhtar En-Naimi, Amami, Boukachour, Person, & Bertelle, 2012; Lavendelis & Grundspenkis, 2009; and Zouhair et al., 2012). Although these are for the most part prototypes, they serve as useful demonstrations of the general approach.

Multiagent architectures depend on a shared agent communication language (ACL) such as Knowledge Query and Manipulation Language (Finin, Fritzson, McKay, & McEntire, 1994), FIPA-ACL (O'Brien & Nicol, 1998), or JADE (Bellifemine, Caire, Poggi, & Rimassa, 2008), all of which are based on speech act theory (Austin, 1965; Searle, 1969). The ACL, combined with a shared ontology (semantic concepts, relationships and constraints), allows the agents to exchange information, to request the performance of a task, and, in certain cases—such as when one agent requests access to restricted data—to deny such requests (Chaib-draa & Dignum, 2002; Kone, Shimazu, & Nakajima, 2000). A multiagent architecture therefore consists of a distributed “society” of agents (Bittencourt et al., 2007), each with its own agenda, semantically-organized knowledge base, and ability to send and receive messages. The messages take the form of speech acts, including *requests*, *directives*, *assertions*, and so forth. Here is an example:

```
request
:receiver pedagogical agent
:sender NLP agent
:ontology electronics
:content (define, capacitor)
```

where the message is clearly identified as a *request*, the receiver is a pedagogical agent, and the sender is a natural language processing (NLP) agent that translates utterances from human language into messages the pedagogical agent can understand. In this case the pedagogical agent can fulfill the request because it has access to an ontology in the domain of electronics, and “knows” how to extract a definition from it, by following an algorithm or production rule. Here’s another example:

```
tell
:receiver pedagogical agent
:sender emotion sensor
:ontology learner affect
:content (learner, confused)
```

where the receiver is again a pedagogical agent, but in this case the sender is an emotion sensing agent reporting its belief that the learner is currently confused. Again, the pedagogical agent can process the contents of the message because it has access to a “learner affect” ontology. As a final example, consider the following:

```
tell
:receiver LMS agent
:sender pedagogical agent
```

```
:ontology learningExperiences
:content (learner, "passed", "helicopter simulation training")
```

where in this case the pedagogical agent is the sender, and the receiver is an LMS agent, which is being told that a certain learner has passed a training course.

These simple examples illustrate several important principles regarding the nature and behavior of multiagent systems. First, note that all three of the software agents are capable of autonomous action, in accordance with their own agendas, and without the need for supervision. The pedagogical agent need not ask the emotion sensor to report its estimate of the learner's affective state. Rather, the emotion sensor reports its beliefs automatically and autonomously, as it does for any agent that has subscribed to its services. Similarly, when it has judged that a learner has passed a course, the pedagogical agent informs the LMS agent, again without having to be asked, simply because the LMS agent has subscribed to its services.

These agents are "lightweight" in the sense that their power lies in their ability to exchange messages with other agents, and to process the contents of these messages based on ontologies that are shared with the agents they exchange messages with, but not necessarily by all of the agents in the system. For example, the NLP agent and pedagogical agent must both have access to the *electronics* ontology, and the LMS agent and pedagogical agent must both share the ontology of *learner experiences*, but neither the emotion sensing agent nor the LMS agent need to know anything about electronics.

Note also that, assuming that the agents' messages are sent over the Internet, all four agents (including the learner) can be at different, arbitrary locations, whether on servers or local devices. Also, any agent can be replaced by any other agent that performs the same function and uses a compatible ACL and associated ontology. If an emotion-sensing agent comes along that does a better job than the original, then, so long as it reads and writes the same kinds of messages and has a compatible ontology (e.g., terms can be translated meaningfully from one ontology to the other), the other agents don't need to be reconfigured in any way. Most importantly, the functionality and value of membership in the society for all participants can increase incrementally, perhaps even dramatically, by registering new agents with new capabilities, or by upgrading the capabilities of the existing members.

Transforming a monolithic ITS legacy system into one with a distributed, multiagent architecture requires two steps: breaking apart existing components into agents and developing ACLs with ITS-tailored ontologies. By encouraging ITS developers to reorganize their systems as services, the Generalized Framework for Intelligent Tutoring (GIFT) provides strong support for this process (Sottilare, Goldberg, Brawner, & Holden, 2012).

2 Criteria for a MAS ITS Framework

Before discussing GIFT specifically, general criteria required for an effective multiagent ITS framework will be discussed. To understand the criteria for a development framework, one must understand something about the stakeholders involved. In this case, as we are focusing on the software development practices of an ITS, these stakeholders are the research groups that develop these systems. So then, what do such groups look like? A recently completed systematic literature review of papers including the terms “intelligent tutoring system” or “intelligent tutoring systems” found that the majority of ITS research was split between two types: major ITS families (those with 10 or more papers in a 4-year period) and single-publication projects (Nye, 2013). Together, these account for over 70% of ITS research with each accounting for a fairly equal share. This means two things. First, any generalized framework should be able to accommodate major ITS projects that have a large prior investment in tools. Second, it means that such a framework should also embrace contributions from new developers who are often focused heavily on only a single ITS component (e.g., linguistic analysis, assessment of learning, haptic interfaces). So then, an ideal framework would facilitate breaking down legacy architectures into multiagent systems and would also make it easy for one-off developers to add or replace a single component. The framework should also not be locked-in to a single template for the components included in the system: not all systems can be easily broken down into the same components. However, this walks a fine line: too much structure hinders innovative designs, while too little structure offers little advantage over a generic architecture (e.g., off-the-shelf service-composition frameworks).

Accommodating these different ends of the spectrum requires a lightweight and flexible architecture. However, what do we mean by “lightweight?” There are multiple meanings for a “lightweight framework” and most of them are favorable in this context. The following features can be either lightweight or heavy: (1) hardware requirements, (2) software expertise to design services, (3) software expertise to use existing services, (4) software expertise to stand up the message-passing layer between agents, and (5) minimal working message ontology. The first requirement is that the no special hardware or excessive computational overhead should be required to use the framework. The computational requirements should be light, rather than imposing heavy overhead or unnecessary inter-process or remote service calls. Components requiring significant setup or maintenance (e.g., databases, web-servers) should be optional or, at a minimum, streamlined with default setups that work out of the box.

Assuming self-interest, for both types of developers (veterans and newcomers), the cost of designing or redesigning for the framework would need to be exceeded by the benefits. This means minimizing development overhead to create new services or refit old services for the framework. The generalized framework would need to allow easy wrapping or replacement of existing designs, rather than forcing developers to maintain two parallel versions of their ITS. Researchers and developers are unlikely to develop for a framework that requires extensive additional work to integrate with. This means that new developers should need to know only the minimal amount of information about the framework in order to integrate with it. There should be little to no work to create a simple service that can interoperate with the framework and default wrappers should exist for multiple programming languages to parse raw messages into native objects. Such wrappers or premade interfaces would allow even relatively “heavy” communication between agents, while keeping developers from needing to know these protocols.

The framework must also make it easy to take advantage of services that others have implemented, such as through a repository of publically-available services. At minimum, it should be significantly easier to use existing services than it is to add a module to the system. This means that the minimal use case (e.g., the "Hello world" case) for the system should be very simple. For example, a single installed package should make it possible to author (or copy) a single text file configuring the system to create a basic ITS test system. Anything required to run a basic example beyond these requirements indicates a "heavier" setup requirement to begin using the framework. If this part of the framework is heavy, first-time ITS authors would be unlikely to use the framework. Moreover, without such ease-of-use, established ITS developers would be unlikely to rework their code to fit such a framework unless they were compensated for these efforts. In the long term, the success and survival of a general framework for tutoring relies on its ability to contribute back to the ITS community. If researchers and developers benefit by reusing services in the system, they will use it. Otherwise, it will fall into obscurity.

Standing up message passing coordination must be lightweight as well. This means that developers should need to expend minimal effort to invoke a layer capable of exchanging messages between services. As such, this layer should have a strong set of defaults to handle common cases and should work out of the box. Additionally, it should be possible to invoke this layer as part of a standalone application (message passing in a single process) or as a remote web service. Consideration must also be given to mobile devices, as mobile applications have specific limitations with respect to their installation, sandboxing (access to other applications), and data transmission.

Finally, agent communication relies on specific messaging languages codified explicitly or implicitly. Three major paradigms are possible to control this communication. The oldest and most traditional paradigm defines API function interfaces for various types of agents or agent functionality, where "messages" are technically function calls on agents. This approach, however, is fragile and better-suited for synchronous local communication than for asynchronous distributed agents. The second paradigm is to define a centrally-defined ontology of messages, which each having an agreed-upon meaning. The main advantage of this system is that it imposes consistency: all agents can communicate using this predefined ontology. However, agreeing upon a specific ontology of messages is an extremely hard task in practice. This approach is "heavy" from the perspective of learning and being constrained by the ontology. The ultimate goal of a shared and stable ontology for ITS is valuable, but offers formidable pragmatic challenges. The third paradigm allows ad-hoc ontologies of messages. At face value, this approach seems flimsy: the ontology of messages is not defined by the agent communication language and services can define their own messages that may not be meaningful to other services as a result. However, this approach is actually fairly popular in research on agent communication languages (Li & Kokar, 2013) and in recent standards bodies, such as the Tin Can API associated with SCORM (Poltrack, Hruska, Johnson, & Haag, 2012). These approaches standardize the format of messages (e.g., how they are structured) but not the content. Instead, certain recommendations for tags and messages are presented but not required. This approach is lightweight: only a small ontology is required and developers are free to extend it.

Lightweight ad-hoc message ontologies show the most promise for an ITS framework using agent message passing. By standardizing the message format, any two services can syntactically understand any message passed to it. However, it allows developers to choose any set of messages for their agent communication language. While in theory this could lead to a Babylon of disjointed ontologies, in practice developers will typically attempt to use established formats for messages first, if they are available. Much like the original design of a computer keyboard or choice of which side of the road to drive on, the starting ontology for a framework can provide a powerful self-reinforcing norm that guides influences work. As such, it is possible to define a core set of suggested messages that are used by the initial set of agents designed for the framework. Additional messages could then be added to the "common core ontology" of messages when they became common practice among new agents added to the service.

3 The GIFT Framework as a MAS ITS

Given these five characteristics, we now look at how well the GIFT architecture matches them in its current form. First, it must be noted that the intentions of the GIFT project are both ambitious and admirable: without the general shift toward service-oriented design for ITSs there would be little value in discussing multiagent ITSs that build upon service-oriented principles. However, this analysis finds that the current implementation of GIFT appears heavier than would be ideal for the needs and practices of ITS developers. This does not mean that GIFT is a bad architecture, simply that it is an architecture that is geared toward the needs of stakeholders other than existing ITS developers (e.g., end-users, sponsors, etc). A great deal of emphasis is placed on reliability and stability, which is more reflective of enterprise use rather than rapid development. The current GIFT implementation implies a "consume and curate" service model rather than a "collaborative repository" service model. With help from GIFT experts, it is certainly possible to integrate tutoring services with GIFT and deliver this tutoring effectively using the architecture. However, the architecture does not seem light enough to allow researchers to build it into their own toolchain. This section first examines the strengths of GIFT as a generalized framework for developing tutoring systems and then considers limitations that might be addressed by future releases.

By far, the primary advantage over existing systems is its dedication to service-oriented principles and modular design. GIFT is the first serious attempt to develop a platform intended to inject a common suite of tutoring services into a variety of applications, including web applications and 3D games (Sottilare, Goldberg, Brawner, & Holden, 2012). GIFT also has a strong commitment to standards-based communication protocols, supporting the Java Messaging Service (JMS) for service communication. Finally, GIFT was developed in Java so it can be efficiently interpreted on web servers and has strong cross-platform capabilities. The hardware requirements for the core GIFT system are also light. Modern systems should have no trouble running the GIFT services and communication layer. Overall, GIFT appears to be well-optimized for efficient delivery and hosting of tutoring web services.

However, the current GIFT implementation has significant limitations as a development framework for tutoring systems. First, the current implementation does not offer an easy road for standing up a minimal working example using the GIFT framework. Installing and setting up the core framework for use is a multi-step process with multiple stages and some third-party dependencies. Running the framework also requires setting up a dedicated database, which could never be considered a light feature. While some GIFT ITS may benefit from such a database (e.g., those hosting surveys), many prototype ITS might make do with simpler triple-stores, serial data (e.g., delimited text files), or even no persistent data storage. Additionally, setting up the GIFT framework does not differentiate between the core architecture meant to handle communication between services versus the services that are bundled with the architecture. A barebones version might remedy this limitation. Services also communicate using a classical API paradigm, which does not offer much flexibility compared to a more explicit message-passing approach. This means that a developer would need to inspect individual service interfaces to figure out the appropriate accessors. Effectively, this locks developers into an ontology of how services should *act* (i.e., remote API requests) rather than what they should *know* (e.g., generic beliefs or knowledge). While this may seem like a subtle difference, a service that only needs to broadcast its knowledge can sidestep designing who receives that information and how it should be used. Finally, GIFT lacks service stubs or wrappers in common languages (e.g., Java, Python, C#) that would make it easy to develop a service that conforms to the framework.

Overall, deploying the GIFT architecture and attempting to develop a new service for the system are both heavy tasks rather than lightweight ones. Without support from the GIFT project, this would make developing for the framework quite costly. The software expertise to design services is heavy, since there are few tools to make this process easier. Despite using a service-oriented paradigm, the system does not offer a suite of example services or stub service in common programming languages. Unless developers have expertise in Java and can carefully inspect the available API, they would not be able to integrate a new service into GIFT. The software expertise to use existing services is also heavy. The minimal use-case example currently installs all GIFT services and requires a database. Services are not handled using a repository or package manager approach, but are simply installed with no streamlined method to manage them. Since there is no way to install the service communication layer as a standalone system, the software expertise to stand up any message-passing layer between agents is also heavy. Finally, no message ontology is available because the system messages are invoked to carry API calls between services. While ontologies for GIFT have been discussed, these ontologies are focused on the types of services in the system rather than the types of messages employed (Sottilare, 2012). This forces communication between services to revolve around the API of services rather than on the information they are passing.

In its current form, the GIFT framework would not be well-suited for a multiagent system ITS. It also does not support many of the aspects of such a framework that would aid either of the major classes of ITS developers to base their projects on GIFT. A one-off innovation, such as a PhD candidate's thesis project, would likely be significantly burdened by the effort to stand up the system without help and would need to learn the API for existing services before they could be useful. A large group focusing on an established ITS architecture would be limited by these factors and also by the lack of interfaces and supporting tools for the programming languages used by their legacy projects. Most importantly, since services do not communicate using a more general agent communication language, significant effort will likely be required to tailor communication to the specific API function interfaces. Without the ability to specify a common message ontology for the agent communication language, it would be impractical to develop a multiagent tutoring system using GIFT. Traditional API's based on interfaces are not well-suited to this task, as they conflate process names with the meanings of the data they produce. Traditional API functions are also poorly suited for dynamic function binding and other advanced patterns that could be used by message-passing agents.

4 Discussion and Recommendations

This analysis has explored the potential benefits and requirements related to building an intelligent tutoring system based on multiagent architecture principles and an agent communication language. These requirements were then compared with the GIFT framework's current capabilities. Our finding is that the current implementation of GIFT is not currently well-suited to these advanced design patterns. While hardware requirements are low, software expertise to design new GIFT services and to use the existing GIFT services is fairly high. Additionally, message system of GIFT currently reflects an API pattern with heavy reliance on knowing the other services in the framework. This is unfortunate, as lighter publish-and-subscribe patterns have become increasingly popular in the industry due to their adaptability (Jokela, Zahemszky, Rothenberg, Arianfar, & Nikander, 2009). This said, GIFT represents a project that is far closer to these patterns than any prior ITS project. GIFT has also spurred discussion on patterns for service-oriented tutoring that were not previously at the forefront of ITS design.

Based on this analysis of GIFT, some design recommendations are indicated for future iterations. From the perspective of developing tutoring agents, the first major recommendation is to center communication of services around explicit message passing where agents publish their knowledge using speech acts. To support this goal, feedback should be gathered from major ITS research groups to propose messages for an initial ontology of recommended messages that determine the information passed between components of the system. To add services to the GIFT framework, developers should only need to know this ontology of messages so they can use it or extend it accordingly. Services should not need to know who their messages are received by, only what messages they receive, what messages they produce, and when they wish to produce a message.

The second major recommendation is the need to separate the GIFT services from the GIFT communication layer. If GIFT is truly a general framework, it must ultimately provide a specialized communication layer as its core. Other services should be treated as plug-ins that can be installed or removed using a package-management approach. This includes the core GIFT services that are bundled with the system. Separating the services from the core architecture would greatly simply the ability to provide a minimal working example and would make the system more flexible overall. As the system itself appears to be designed with such boundaries in mind, this should primarily be a matter of how setup packages are structured and installed.

Related to this issue, a very basic installation that works “out of the box” must be available for developers to start working with GIFT.

The third major recommendation is that GIFT should provide small suites of utilities, wrappers, and stubs to help develop services using a variety of common languages. A generalized system must not assume that developers will convert their code to Java or build their own communication wrappers for their native language. While the use of remote procedure API calls has sidestepped this issue slightly, it has not completely removed it. Additionally, a more flexible message-passing paradigm would require such supporting tools to an even greater extent.

Finally, in the present analysis we have focused only on issues of system architecture, as is proper given that GIFT intends to serve as a general purpose framework, not a stand-alone ITS. However, in so doing we have arguably paid insufficient attention to other important issues that GIFT approaches, such as the need for shareable domain models, learner models, and instructional modules. As the developers of GIFT have pointed out, legacy ITSs tend to be built as “unique, one-of-a-kind, largely domain-dependent solutions focused on a single pedagogical strategy” (Sottilare, Brawner, Goldberg & Holden, 2012:1). After some four decades of independent effort, a case can be made that the time has come for a much greater degree of collaboration and sharing among members of the ITS community, including both veterans and newcomers. This means not just the sharing of ideas, but of working software objects and structures. The development of a lightweight, multiagent architecture that supports “autonomous cooperation” among communities of distributed software agents united by an emergent common language offers a first step in the process, but it is by no means the last.

5 References

1. Austin, J. L.: How to do things with words. Oxford University Press, New York (1965)
2. Bellifemine, F., Caire, G., Poggi, A., Rimassa, G.: JADE: A software framework for developing multi-agent applications. Lessons learned, Information and Software Technology, 50(1), 10–21 (2008)
3. Bittencourt, I. I., de Barros Costa, E., Almeida, H. D., Fonseca, B., Maia, G., Calado, I., Silva, A. D.: Towards an ontology-based framework for building multiagent intelligent tutoring systems. In: Simpósio Brasileiro de Engenharia De Software. Workshop on Software Engineering for Agent-oriented Systems, III, João Pessoa, 2007. Proceedings of the Porto Alegre, SBC, pp. 53–64 (2007)
4. Chaib-draa, B., Dignum, F.: Trends in agent communication language. Computational Intelligence, 18(2), 89–101 (2002)
5. Chen, W., Mizoguchi, R.: Learner model ontology and learner model agent. Cognitive Support for Learning-Imagining the Unknown, 189–200 (2004)
6. El Mokhtar En-Naimi, A. Z., Amami, B., Boukachour, H., Person, P., Bertelle, C.: Intelligent Tutoring Systems Based on the Multi-Agent Systems (ITS-MAS): The Dynamic and Incremental Case-Based Reasoning (DICBR) Paradigm. IJCSI International Journal of Computer Science Issues 9(6), 112–121 (2012)

7. Finin, T., Fritzson, R., McKay, D., McEntire, R.: KQML as an agent communication language. In: Proceedings of the third international conference on Information and knowledge management, pp. 456–463. ACM (1994, November)
8. Franklin, S., Graesser, A.: Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In: Intelligent agents III agent theories, architectures, and languages, pp. 21–35. Springer Berlin Heidelberg (1997)
9. Hülsmann, M., Scholz-Reiter, B., Freitag, M., Wucisk, C., De Beer, C.: Autonomous cooperation as a method to cope with complexity and dynamics?–A simulation based analyses and measurement concept approach. In: Y. Bar-Yam (ed.), Proceedings of the International Conference on Complex Systems (ICCS 2006), vol. 2006. Boston, MA, USA (2006)
10. Jokela, P., Zahemszky, A., Rothenberg, C., Arianfar, S., Nikander, P.: LIPSIN: Line speed publish/subscribe inter-networking. In: ACM SIGCOMM Computer Communication Review, 39(4), pp. 195–206. ACM Press (2009, August)
11. Kone, M. T., Shimazu, A., Nakajima, T.: The state of the art in agent communication languages. Knowledge and Information Systems, 2(3), 259–284 (2000)
12. Lavendelis, E., Grundspenkis, J.: Design of multi-agent based intelligent tutoring systems. Scientific Journal of Riga Technical University. Computer Sciences, 38(38), 48–59 (2009)
13. Li, S., Kokar, M. M.: Agent Communication Language. In: Flexible Adaptation in Cognitive Radios, pp. 37–44. Springer New York (2013)
14. Nye, B. D.: ITS and the Digital Divide: Trends, Challenges, and Opportunities. In: Artificial Intelligence in Education (In Press).
15. O'Brien, P. D., Nicol, R. C.: FIPA–towards a standard for software agents. BT Technology Journal, 16(3), 51–59 (1998)
16. Poltrack, J., Hruska, N., Johnson, A., Haag, J.: The Next Generation of SCORM: Innovation for the Global Force. In: The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), vol. 2012, no. 1. National Training Systems Association (2012, January)
17. Searle, J. R.: Speech acts: An essay in the philosophy of language. Cambridge University Press (1969)
18. Sottilare, R. A.: Making a case for machine perception of trainee affect to aid learning and performance in embedded virtual simulations. In: Proceedings of the NATO HFM-169 Research Workshop on the Human Dimensions of Embedded Virtual Simulation. Orlando, Florida (2009, October)
19. Sottilare, R. A.: Considerations in the development of an ontology for a generalized intelligent framework for tutoring. In: Proceedings of the International Defense and Homeland Security Simulation Workshop (2012)
20. Sottilare, R. A., Goldberg, B. S., Brawner, K. W., Holden, H. K.: A Modular Framework to Support the Authoring and Assessment of Adaptive Computer-Based Tutoring Systems (CBTS). In: The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), vol. 2012, no. 1. National Training Systems Association (2012, January)
21. Windt, K., Böse, F., Philipp, T.: Criteria and application of autonomous cooperating logistic processes. In: Gao, J.X., Baxter, D.I., Sackett, P.J. (eds.) Proceedings of the 3rd International Conference on Manufacturing Research. Advances in Manufacturing Technology and Management (2005)
22. Zouhair, A., En-Naimi, E. M., Amami, B., Boukachour, H., Person, P., Bertelle, C.: Intelligent tutoring systems founded on the multi-agent incremental dynamic case based reasoning. In: Information Science and Technology (CIST), 2012 Colloquium, pp. 74–79. IEEE (2012, October)

Authors

Benjamin D. Nye is a post-doctoral fellow at the University of Memphis, working on tutoring systems architectures as part of the ONR STEM Grand Challenge. Ben received his Ph.D. from the University of Pennsylvania and is interested in ITS architectures, educational technology for development, and cognitive agents.

Dr. Chip Morrison is a Faculty Affiliate at IIS. A graduate of Dartmouth, Dr. Morrison holds an M.A. from the University of Hong Kong and an Ed.D. from Harvard. His current research interests include models of human cognition and learning, and the application of these models to conversation-based intelligent learning systems.

Recommendations For The Generalized Intelligent Framework for Tutoring Based On The Development Of The DeepTutor Tutoring Service

VASILE RUS, NOBAL NIRAULA, MIHAI LINTEAN, RAJENDRA BANJADE, DAN STEFANESCU, WILLIAM BAGGETT
The University of Memphis
Department of Computer Science/Institute for Intelligent Systems
Memphis, TN 38138
vrus@memphis.edu

Abstract. We present in this paper the design of DeepTutor, the first dialogue-based intelligent tutoring system based on Learning Progressions, and its implications for developing the Generalized Framework for Intelligent Tutoring. We also present the design of SEMILAR, a semantic similarity toolkit, that helps researchers investigate and author semantic similarity models for evaluating natural language student inputs in conversatioanl ITSs. DeepTutor has been developed as a web service while SEMILAR is a Java library. Based on our experience with developing DeepTutor and SEMILAR, we contrast three different models for developing a standardized architecture for intelligent tutoring systems: (1) a single-entry web service coupled with XML protocols for queries and data, (2) a bundle of web services, and (3) library-API. Based on the analysis of the three models, recommendations are provided.

Keywords: intelligent tutoring systems, computer based tutors, dialogue systems

1 Introduction

The General Framework for Intelligent Tutoring (GIFT; Sottilare et al, 2012) aims at creating a modular ITS/CBTS (intelligent tutoring systems/computer-based tutoring systems) framework and standards to foster “reuse, support authoring and optimization of CBTS strategies for learning, and lower the cost and skillset needed for users to adopt CBTS solutions for military training and education.” GIFT has three primary functions: (1) to help with developing components for CBTS and whole tutoring systems; (2) to provide an instructional manager that integrates effective and exploratory tutoring principles and strategies for use in CBTS; and (3) to provide an experimental test bed to analyze the effectiveness and impact of CBTS components, tools, and methods. That is, GIFT is both a software environment and standardization effort. The availability of a GIFT software package suggests that for now the software environment has been given priority to standardization efforts. This paper intends to help make progress towards a GIFT standardization.

To that end, we present the design of DeepTutor (www.deeptutor.org; Rus et al., to appear), the first CBTS based on the emerging framework of Learning Progressions proposed by the science education research community (LPs; Corcoran, Mosher, & Rogat, 2009). LPs can be viewed as incrementally more sophisticated ways to think about an idea that emerge naturally while students move toward expert-level understanding of the idea (Duschl et al., 2007). That is, LPs capture the natural sequence of mental models and mental model shifts students go through while mastering a topic. It is this learner-centric view that differentiates LPs from previous attempts to reform science education. The LPs framework provides a promising way to organize and align content, instruction, and assessment strategies in order to give students the opportunity to develop deep and integrated understanding of science ideas.

DeepTutor is developed as a web service and a first prototype is fully accessible through a browser from any Internet-connected device, including regular desktop computers and mobile devices such as tablets. As of this writing, DeepTutor is designed as a bundle of two web services: (1) the tutoring service itself accessed by learners, and (2) the support service which includes everything else: authoring and content management, experiment management, user management, and instruction management. The latter service is viewed as a single service because there is a single-entry point to access all these functions. The tutoring service exports its functionality through an XML-based protocol. Third party developers can use their own development environments to design custom DeepTutor clients and integrate them with the DeepTutor tutoring service; all they need is to understand and generate an XML-like protocol, which is a query-language for accessing DeepTutor functionality.

We contrast the DeepTutor design with the design of another software environment, SEMILAR (www.semanticsimilarity.org; Rus et al., 2013). SEMILAR can be used to author semantic similarity methods for semantic processing tasks such as the task of assessing students' natural language inputs in dialogue-based CBTSs. SEMILAR, a SEMantic simILARity toolkit, has been designed as a Java library. Access to SEMILAR functionality is already available through a Java API (Application Programming Interface). Users can use the semantic similarity methods in SEMILAR as long as they link the SEMILAR library to their own Java programs. If a developer were to use SEMILAR from non-Java applications, a solution would be for the SEMILAR library to export its functionality through an XML-like protocol which is easily readable from any programming language. This latter integration solution is basically the export of functionality approach available in the DeepTutor tutoring service. SEMILAR has not been developed as a web service because it was initially developed for our own internal use. We have plans to make it available as a web service in the future. A GUI-based Java application has been developed and is currently tested to offer non-programmers easy access to the SEMILAR functionality.

The two designs, DeepTutor and SEMILAR, will help us discuss concretely three models for standardizing and implementing CBTS functionality to meet GIFT's goals: (1) a single-entry web service, e.g. the two DeepTutor services can be collated into one service (a one-stop-shop model); (2) a bundle of web services – the current DeepTutor design in which different functionality is accessed through different service points, and (3) a library of components accessed through an API. The three models share the common requirement of standardizing the communication between a client/user and provider of tutoring components/functions. While all three models have advantages and disadvantages, we favor the web services models for a Generalized Framework for Intelligent Tutoring as these models better suit the emerging world of mobile computing in which users access services in the cloud over the network as opposed to downloading full applications on their local, energy-sensitive mobile devices. Furthermore, the combination of a tutoring service and XML-based protocols for data and commands/queries fits very well with recent standards for representing knowledge proposed by the Semantic Web community, standards for authoring behavior of dialogue systems (see the FLORENCE dialogue manager framework; Fabbrizio & Lewis, 2004), or previous work in the intelligent tutoring community (see CircSim's mark-up language; Freedman et al., 1998).

The rest of the paper is organized as in the followings. The next section provides an overview of the DeepTutor web service. Then, we describe the design of the SEMILAR library. We conclude the paper with Discussion and Conclusions in which we make recommendations for GIFT based on the three models we discussed.

2 The Intelligent Tutoring Web Service DeepTutor

DeepTutor is a conversational ITS that is intended to increase the effectiveness of conversational ITSs beyond the interactivity plateau (VanLehn, 2011) by promoting deep learning of complex science topics through a combination of advanced domain modeling methods (based on LPs), deep language and discourse processing algorithms, and advanced tutorial strategies. DeepTutor currently targets the domain of conceptual Newtonian Physics but it is designed with scalability in mind (cross-topic, cross-domain).

DeepTutor is a problem solving coaching tutor. DeepTutor challenges students to solve problems, called tasks, and scaffolds their deep understanding of complex scientific topics through constructivist dialogue and other elements, e.g. multimedia items. DeepTutor uses the framework of Learning Progressions (LPs) to drive its scaffolding at macro- and micro-level (Rus et al, to appear). There is an interesting interplay among assessment, LPs, instructional tasks, and advanced tutoring strategies that is finely orchestrated by DeepTutor. The LPs are aligned with an initial, pre-tutoring assessment instrument (i.e., pretest) which students must complete before interacting with the system. Based on this first summative assessment, an initial map of students' knowledge level with respect to a topic LP is generated. The LPs encode both knowledge about the domain and knowledge about students' thinking in the form of models that students use to reason about the domain. The student models vary from naïve to weak to strong/mastery models. For each level of understanding in the LP a set of instructional tasks are triggered that are deemed to best help students make progress towards mastery, which coincides with the highest level of understanding modeled by the LP.

The task representation is completely separated from the executable code and therefore DeepTutor is compliant with the principles adopted by GIFT from Patil and Abraham (2010). Also, in accordance with GIFT principles (Sottilare et al., 2012), DeepTutor's pedagogical module interacts with the learner module (the Student) and adapts the scaffolding tasks and dialogue according to the learner's level of knowledge.

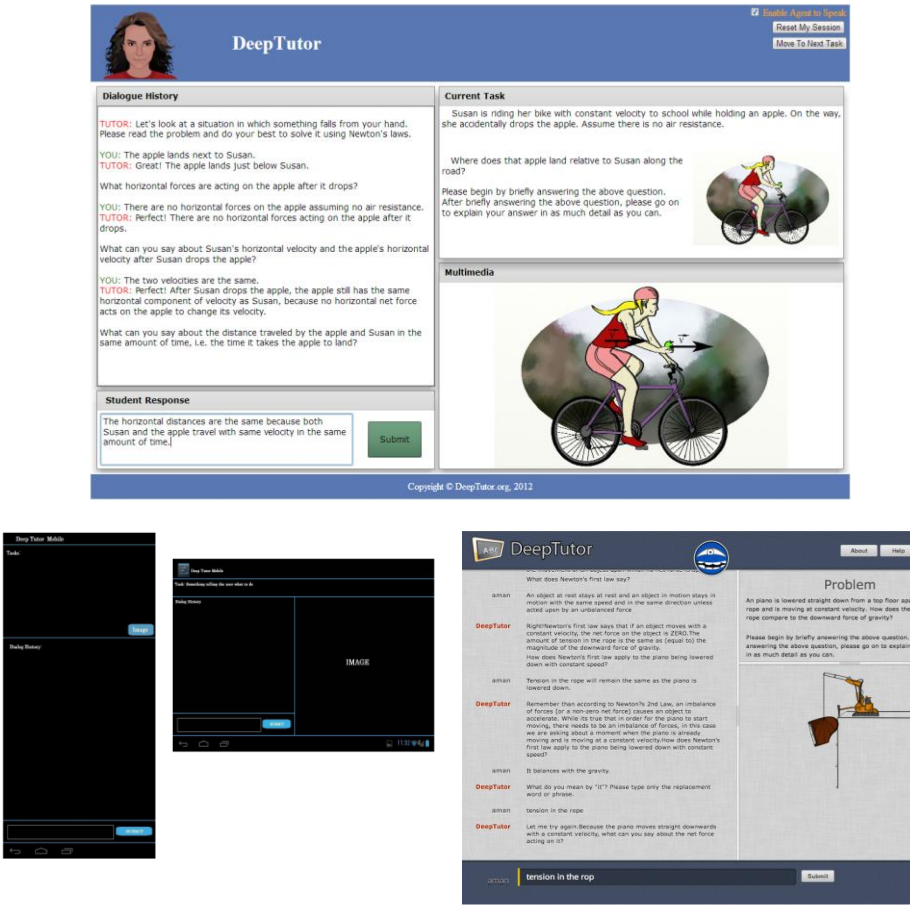
DeepTutor is an ongoing project. As of this writing, different modules are at different stages of maturity. For instance, our LP has been empirically validated based on data collected from 444 high-school student responses. Other components, e.g. the general knowledge module that can handle tasks related to general knowledge such as answering definitional questions ("What does *articulate* mean?"), is still in the works. The system as a whole will be fully validated in the next 6-12 months.

As already mentioned, DeepTutor has been designed as a web service accessible via HTML5-compatible clients, typically web browsers. The familiarity of users with web browsers and eliminating the need to install software packages (except the web browser) on each user's own computer environment makes it extremely convenient for users to access DeepTutor from any Internet-connected device and at the same time opens up unprecedented economies of scale for tutoring research. For instance, during Spring 2013 DeepTutor has been successfully used by more than 300 high-school students[7] from their Internet-device of choice (outside of traditional classroom instruction or experimental lab): home computer, tablet, mobile phones, or library computer.

All communication between the client and the DeepTutor server is handled through an XML-like protocol. The protocol specifies both commands and data that both client and server can interpret. The client communicates user actions and data to the server and the server replies with appropriate responses. Currently, the responses are in the form of display commands and values for various tutoring elements that are visible to the user on screen. That is, the client simply uses the information to update the corresponding interface elements, e.g. the client needs to update the dialogue history box with the most recent DeepTutor feedback response. The protocol contains sufficient information for learner software clients to display the elements of the standard DeepTutor interface. At the same time, the client uses the XML protocol to send the DeepTutor server important information about the user, e.g. user actions such as turning the talking head off, typed responses, time stamps, etc.

There are two major phases for learner clients to connect to the full DeepTutor system: the user authentication and initialization phase and the tutoring phase. In the authentication and initialization phase the user authenticates herself. A set of initialization parameters are sent to the DeepTutor system as well. Currently, the initialization parameters are set from the instructor view of the system, e.g. the researcher/experimenter or instructor/teacher can set a particular instructional strategy to be used by the system for a particular user or groups of learners. We can imagine in the future that these parameters are set dynamically based on the student model retrieved from a persistent database of learner information.

[7] This group of students is different from the 444 student group used for validating the LP.



TUTOR: Let's look at a situation in which something falls from your hand. Please read the problem and do your best to solve it using Newton's laws.

YOU: The apple lands next to Susan.
TUTOR: Great! The apple lands just below Susan.

What horizontal forces are acting on the apple after it drops?

YOU: There are no horizontal forces on the apple assuming no air resistance.
TUTOR: Perfect! There are no horizontal forces acting on the apple after it drops.

What can you say about Susan's horizontal velocity and the apple's horizontal velocity after Susan drops the apple?

YOU: The two velocities are the same.
TUTOR: Perfect! After Susan drops the apple, the apple still has the same horizontal component of velocity as Susan, because no horizontal net force acts on the apple to change its velocity.

What can you say about the distance traveled by the apple and Susan in the same amount of time, i.e. the time it takes the apple to land?

Susan is riding her bike with constant velocity to school while holding an apple. On the way, she accidentally drops the apple. Assume there is no air resistance.

Where does that apple land relative to Susan along the road?

Please begin by briefly answering the above question. After briefly answering the above question, please go on to explain your answer in as much detail as you can.

The horizontal distances are the same because both Susan and the apple travel with same velocity in the same amount of time.

What does Newton's first law say?

aman An object at rest stays at rest and an object in motion stays in motion with the same speed and in the same direction unless acted upon by an unbalanced force

DeepTutor Right!Newton's first law says that if an object moves with a constant velocity, the net force on the object is ZERO.The amount of tension in the rope is the same as (equal to) the magnitude of the downward force of gravity.
How does Newton's first law apply to the piano being lowered down with constant speed?

aman Tension in the rope will remain the same as the piano is lowered down.

DeepTutor Remember than according to Newton?s 2nd Law, an imbalance of forces (or a non-zero net force) causes an object to accelerate. While its true that in order for the piano to start moving, there needs to be an imbalance of forces, in this case we are asking about a moment when the piano is already moving and is moving at a constant velocity.How does Newton's first law apply to the piano being lowered down with constant speed?

aman It balances with the gravity.

DeepTutor What do you mean by "it"? Please type only the replacement word or phrase.

aman tension in the rope

DeepTutor Let me try again.Because the piano moves straight downwards with a constant velocity, what can you say about the net force acting on it?

aman tension in the rop

An piano is lowered straight down from a top floor apa
rope and is moving at constant velocity. How does the
rope compare to the downward force of gravity?

Please begin by briefly answering the above question.
answering the above question, please go on to explain
in as much detail as you can.

Figure 10. Three DeepTutor clients showing three different renderings of the learner-view of the DeepTutor Service: the currently official learner view in DeepTutor (top), an under-development Android app (bottom left) and a client developed for a Masters project (bottom right).

Client applications that access the full DeepTutor tutoring system (not individual components) can be designed quite easily. The main reason is the relatively simple but efficient current interface that allows the learner to focus on the interactive tutorial dialogue. Figure 1 bottom shows on the left-hand side an Android-based app client for DeepTutor designed by a small team of 5 Computer Science undergraduate students as a semester-long class project. The app has an interface design for a vertical versus horizontal positioning of the mobile device. The right-hand side of Figure 1 includes another DeepTutor client designed by a Masters student in Computer Science as his Masters project on Human-Computer Interaction.

It should be noted that more complex learner views are in the plans for DeepTutor. For instance, we plan to add several supplemental instructional aids and monitoring and informing elements such as how many tasks are left to cover in the current session or game-like features such as showing what percentage of a learner’s peers successfully finished the current task. The current interface of DeepTutor is as simple as it can be and it was intentionally kept this way. The goal was to reduce the number of on-screen distractors in order for the learner to focus on the tutorial dialogue. Adding more elements would make the interface richer which could distract the learners from the main tutorial interaction. It would be an interesting topic to investigate though.

We imagine that other users, e.g. developer of tutoring systems, may need to access specific functionality/components of DeepTutor according to the GIFT goals. As an example, we can imagine someone willing to access the output of the assessment module. As of this writing, the client-server protocol does not allow export of specific functionality. To allow export of functionality at a finer-grain level the current DeepTutor XML protocol must be extended such that the server provides developers/researcher clients output from specific modules, e.g. the assessment module. The exact format of the query and response must be clearly defined.

We believe that efforts to standardize access to GIFT-defined CBTS modules using XML protocols are best. The specification of these protocols needs to be done at different levels of abstractness such that the protocol is general enough to be applicable to all types of tutoring systems (at higher, more general levels of specification) and detailed enough for specific types of tutoring systems to be readily implementable by various groups. For instance, a general specification for querying the assessment module would include a general query element that indicates that an user input is needed together with a context variable which may contain other useful information for best assessing the student input (the context variable could be as simple as an user identifier and a session identifier or much more complex including a comprehensive list of factors that might impact assessment) and the format of the response from the assessment component of the tutoring service. This general specification can be further specified for *benchmark-based tutoring systems* (AutoTutor – Graesser et al., 2005, Guru – Olney et al. 2012; DeepTutor – Rus et al., to appear) as well as for *reasoning-based tutoring systems* (Why-Atlas; VanLehn et al., 2007). We use this broad categorization of tutoring systems to help us illustrate the need for further specifying general query formats. A *benchmark-tutoring system* is one that requires an expert-generated or benchmark response against which the student response is assessed (DeepTutor is such a system; Rus et al., to appear). For benchmark-tutoring systems the assessment query will need to pass (a pointer to) the benchmark response as one of the input parameters. *Reasoning-based systems* are able to infer the correct response automatically (Why-Atlas; VanLehn et al., 2007). For reasoning-based systems the benchmark response may not be needed but instead (a pointer to) a knowledge base.

In summary, a web service together with XML-based protocols may offer the best option for moving forward in GIFT. The advantage of using a web service solution with an XML-based protocol has the advantage of being easily extendable (new functionality can be added by simple adding new tags in the XML protocol). Another advantage is the decoupling the logical view from the actual implementation. The decoupling of functionality from actual implementation can be very useful. For example, the XML protocol can offer a GIFT-like view of the system with components so defined to meet GIFT standards while the actual, back-end implementation can be so designed to best fit particular types of ITSs. Sometimes refactoring and exporting functionality is conceptually challenging as for some tutoring systems there is a tight connection between components that GIFT suggest be separate. For instance, in LP-based ITSs such as DeepTutor, there is a tight relationship between learner models and the domain model because the domain is organized from a learner perspective (Rus et al., in press). Separating the learner model from the domain model is conceptually challenging and probably not recommended. The decoupling of functionality allows keeping the best implementation while offering differing views recommended by standards.

The combination of web service/XML protocol is also more advantageous when it comes to updates and extensions. There is no need to download and recompile a client application with the latest version of a component or the whole tutoring system.

We conclude this section by noting that the service model can further be refined into two types of service-based models: single service versus bundle of services. The current DeepTutor system is a **bundle of services**. In this model, the functionality of the various modules would be available as separate web services, e.g. the assessment module could be a separate web service. There are some interesting aspects of the **bundle of services model**. For instance, in DeepTutor some functionality is offered through a combination of the two DeepTutor services: debugging capabilities are offered through a combination of the tutoring and support services. That is, a developer polishing various components has to use both services.

All services can eventually be bundled together in a single, deep service (containing many subservices) in which case we have a **single-entry service model**. This model implements the concept of a one-stop-shop meaning users will use on access point for the components or the whole tutoring system.

3 The SEMILAR Library For Assessing Natural Language Student Inputs

Our SEMILAR (SEMantic similarity) toolkit, includes implementations and extensions of a number of algorithms proposed over the last decade to address the general problem of semantic similarity. SEMILAR includes algorithms based on Latent Semantic Analysis (LSA; Landauer et al., 2007), Latent Dirichlet Allocation (Blei, Ng, & Jordan, 2003), and lexico-syntactic optimization methods with negation handling (Rus & Lintean, 2012a; Rus et al., 2012b); Rus et al, in press). Due to space reasons, we do not present the set of methods available but rather discuss the design of SEMILAR as a Java library and its implications for using an akin design for GIFT.

The Java library design for SEMILAR has the advantage of being easily integrated as compiled code into Java applications which, at least in theory, should be platform independent. However, users have to download the whole package, install it, and then compile it with their tutoring systems. If these systems or components are written in a programming language different from Java, extra effort will be needed for integration. We call this **the library-API model** for a GIFT framework. Indeed, a GIFT framework based on the library-API model will require downloading and installing large software packages on various platforms by users of various technical backgrounds which may make the whole effort more challenging. For instance, the SEMILAR library and application is 300MB large (it includes large models for syntactic parsing among other things). SEMILAR can be regarded as a tutoring component for assessing students' natural language inputs. If ITS developers were to use SEMILAR as a library they have to download it and integrate it in their products. They have to install and update the API when updates become available. In fact, this is how SEMILAR is currently integrated in DeepTutor. Changes in implementation, e.g. bug fixes, would require a new download and reintegration of the systems that rely on the library. When SEMILAR will be available as a web service, all is needed is understanding the API, in the form of an XML-based communication protocol, and connect to the tutoring service. The need for a network connection are a potential risk for the service model in the form of network congestion which may make the service inaccessible or slow at times.

4 Discussion and Conclusions

We presented three models based on our experience with implementing a set of coherent functionalities related to intelligent tutoring systems and semantic processing. Each of the models has its own advantages and disadvantages. Ideally, all three models should be adopted by GIFT. However, if it were to choose we believe that the service-based models are the best solution for an emerging world of mobile devices in which accessing software services in the cloud is becoming the norm. The library-API and web service solutions are functionally equivalent with the former presenting more technical challenges for users with diverse backgrounds and computing environments and also being less suitable for a mobile computing world.

One apparent downside of the web service model is that potential developers cannot alter the code themselves in order to conduct research. This is just an apparent downside as a quick fix would be for each component to offer enough parameters, in the form of a switchboard, to allow potential users to alter behavior without the need to change the code. In fact, this solution should be preferred as users would not need to spend time to understand and alter the code, a tedious and error-prone activity.

Standardization efforts for XML-based protocols may start with previous efforts where available. For instance, the dialogue processing community has made attempts to standardize dialogue acts/speech acts, a major component in dialogue-based ITSs, for more than a decade. The resulting Dialogue Act Mark-Up in Several Layers (DAMSL) XML schema can be used as a start to standardize speech acts in dialogue ITSs.

In summary, we favor a **one-stop-shop service** model with **switchboard**-like facilities for implementing GIFT. Table 1 below illustrates the pros and cons of the three models discussed in this paper.

Table 6. Comparison of the three proposed model: single-entry service, bundle of services, and library.

	One-Stop-Shop/Single-Entry Service	**Bundle of Services**	**Library**
Programming Language Independent	YES	YES	NO
Install and update on local machine/ environment	NO	NO	YES
Fit for emerging mobile and cloud-computing fitness	EXCELLENT	EXCELLENT	POOR
Customization	VERY GOOD	VERY GOOD	EXCELLENT
Cost of Customization	LOW	MEDIUM	HIGH (error prone and time to work with someone else' code)
Extendible	EXCELLENT	EXCELLENT	GOOD

5 Acknowledgements

The authors would like to thank members of DeepTutor project (www.deeptutor.org). This research was supported by the Institute for Education Sciences (IES) under award R305A100875 by a grant from the Army Research Lab (ARL). Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the IES or ARL.

6 References

1. Blei, D.M., Ng, A.Y., & Jordan, M.I. 2003. Latent dirichlet allocation, The Journal of Machine Learning Research 3, 993-1022.
2. Corcoran, T., Mosher, F.A., & Rogat, A. (2009). Learning progressions in science: An evidencebased approach to reform. Consortium for Policy Research in Education Report #RR-63. Philadelphia, PA: Consortium for Policy Research in Education.
3. Duschl, R.A., Schweingruber, H.A., & Shouse, A. (Eds.). (2007). Taking science to school: Learning and teaching science in grades K-8. Washington, DC: National Academy Press.
4. Graesser, A. C.; Olney, A.; Haynes, B. C.; and Chipman, P. 2005. Autotutor: A cognitive system that simulates a tutor that facilitates learning through mixed-initiative dialogue. In Cognitive Systems: Human Cognitive Models in Systems Design. Mahwah: Erlbaum.
5. Landauer, T.; McNamara, D. S.; Dennis, S.; and Kintsch, W. (2007). Handbook of Latent Semantic Analysis. Mahwah, NJ: Erlbaum.
6. Freedman, Reva, Yujian Zhou, Jung Hee Kim, Michael Glass, and Martha W. Evens.

7. SGML-Based Markup as a Step toward Improving Knowledge Acquisition for Text Generation AAAI 1998 Spring Symposium: Applying Machine Learning to Discourse Processing
8. VanLehn, K. (2011). The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems, Educational Psychologist, 46:4, 197-221.
9. Olney, A., D'Mello, A., Person, N., Cade, W., Hays, P., Williams, C., Lehman, B., & Graesser, A. (2012). Guru: A computer tutor that models expert human tutors. In S. Cerri, W. Clancey, G. Papadourakis & K. Panourgia (Eds.), Proceedings of the 11th International Conference on Intelligent Tutoring Systems (pp. 256-261). Springer-Verlag.
10. Patil, A. S., & Abraham, A. (2010). Intelligent and Interactive Web-Based Tutoring System in Engineering Education: Reviews, Perspectives and Development. In F. Xhafa, S. Caballe, A. Abraham, T. Daradoumis, & A. Juan Perez (Eds.), Computational Intelligence for Technology Enhanced Learning. Studies in Computational Intelligence (Vol 273, pp. 79-97). Berlin: Springer-Verlag.
11. Rus, V. & Lintean, M. (2012a). A Comparison of Greedy and Optimal Assessment of Natural Language Student Input Using Word-to-Word Similarity Metrics, Proceedings of the Seventh Workshop on Innovative Use of Natural Language Processing for Building Educational Applications, NAACL-HLT 2012, Montreal, Canada, June 7-8, 2012.
12. Rus, V., Lintean, M., Moldovan, C., Baggett, W., Niraula, N., Morgan, B. (2012b). The SIMILAR Corpus: A Resource to Foster the Qualitative Understanding of Semantic Similarity of Texts, In Semantic Relations II: Enhancing Resources and Applications, The 8th Language Resources and Evaluation Conference (LREC 2012), May 23-25, Instanbul, Turkey.
13. Rus, V.; Lintean, M.; Banjade, R.; Niraula, N.; Stefanescu, D. (2013). SEMILAR: The Semantic Similarity Toolkit, The 51st Annual Meeting of the Association for Computational Linguistics, System Demo Paper, August 4-9, 2013, Sofia, Bulgaria.
14. Rus, V., D'Mello, S., Hu, X., and Graesser, A.C. (to appear) .Recent Advances In Conversational Intelligent Tutoring Systems, AI Magazine.
15. VanLehn, K., Graesser, A. C., Jackson, G. T., Jordan, P., Olney, A., & Rose, C. P. (2007). When are tutorial dialogues more effective than reading? Cognitive Science, 31, 3-62.
16. VanLehn, K. (2011). The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems, Educational Psychologist, 46:4, 197-221.

Authors

Vasile Rus: Dr. Vasile Rus is an Associate Professor of Computer Science with a joint appointment in the Institute for Intelligent Systems (IIS). Dr. Rus' areas of expertise are computational linguistics, artificial intelligence, software engineering, and computer science in general. His research areas of interest include question answering and asking, dialogue-based intelligent tutoring systems (ITSs), knowledge representation and reasoning, information retrieval, and machine learning. For the past 10 years, Dr. Rus has been heavily involved in various dialogue-based ITS projects including systems that tutor students on science topics (DeepTutor), reading strategies (iSTART), writing strategies (W-Pal), and metacognitive skills (MetaTutor). Currently, Dr. Rus leads the development of the first intelligent tutoring system based on learning progressions, DeepTutor (www.deeptutor.org). He has coedited three books, received several Best Paper Awards, and authored more than 90 publications in top, peer-reviewed international conferences and journals. He is currently Associate Editor of the International Journal on Artificial Intelligence Tools.

Nobal Niraula: Nobal B. Niraula received the B.E. in computer engineering from Pulchowk Campus, Tribhuvan University, Nepal, the M.E. in information and communication technology and the M.Sc. in communication networks and services from Asian Institute of Technology, Thailand and Telecom SudParis, France respectively. He was a research engineer at INRIA, Saclay, France where he worked in semantic web, database systems and P2P networks. Currently, he has been doing his PhD at The University of Memphis, USA. His research interests are primarily in Intelligent Tutoring Systems, Dialogue Systems, Information Extraction, Machine Learning, Data Mining, Semantic Web, P2P and Ad hoc networks. He has received the best paper and the best presentation awards. He also has intern experiences in leading research labs such as AT&T Labs Research.Mihai Lintean: Dr. Mihai Lintean is currently a research scientist at Carney Labs LLC and previous to that he was a Postdoctoral Research Fellow in the Computer Science Department at the University of Memphis, where he worked with Dr. Vasile Rus ondialogue based tutoring systems for teaching conceptual physics to high school students. Mihai's primary research interests are in Natural Language Processing (NLP), with focused applicability on educational technologies such as intelligent tutoring systems. Particularly he is interested in measuring semantic similarity between texts, representing knowledge through relational diagrams of concepts, automatic generation of questions, and using various machine learning techniques to solve other complex NLP problems. Mihai has published numerous papers and articles in reputable, peer-reviewed conferences and journals. He currently serves as co-chair of the Applied Natural Language Processing Special Track at the 25th International Conference of the Florida Artificial Intelligence Research Society (FLAIRS 2012).

Rajendra Banjade: Rajendra Banjade is a PhD student in Computer Science at The University of Memphis. He is a research assistant in the DeepTutor project (www.deeptutor.org) - a dialogue based tutoring system. Rajendra's research interests are in the area of Natural Language Processing, Information Retrieval, and Data Mining. Currently, he is focusing on measuring semantic similarity of short texts (word and sentence level) using knowledge based and corpus based methods and heading towards more human like inferencing techniques. His current research focus is on robust methods to evaluate student answers in conversational intelligent tutoring systems. He is keenly dedicated to enhancing the SEMILAR toolkit (www.semanticsimilarity.org) which is an off-the-shelf semantic similarity toolkit. Before joining The University of Memphis, he worked for five years as a Software Engineer (R&D) at Verisk Information Technologies CMMI III, Kathmandu (a subsidiary of Verisk Analytics inc.) where he got opportunities working on various healthcare data mining projects including DxCG Risk Solutions engine. He received an outstanding employee award at Verisk. Rajendra is a certified Scrum Master and Software Developer, and Certified HIPAA professional. He holds bachelor's degree in Computer Engineering.

William B. Baggett: William B. Baggett earned a PhD in Cognitive Psychology from The University of Memphis in 1998. He also holds an MS in Computer Science and an MBA in Management Information Systems. William is currently a Project Coordinator in the Computer Science Department at The University of Memphis, where he works on DeepTutor. DeepTutor is an intelligent tutoring system, implemented as a web application, which uses natural language dialogue to teach conceptual physics to high school and college students. Previously, William was a Professor and part-time Department Chair of Computer Information Systems at Strayer University and an adjunct Professor of Computer Science at The University of Memphis. In both positions, William taught graduate and undergraduate Computer Science courses, mentored, tutored, and advised students, and developed new curricula. He was also a Business Analyst at FedEx Express where he wrote software specifications for PowerPad, a mission-critical handheld computer carried by FedEx Express couriers. PowerPad software is designed to promote optimal courier behavior including the efficient pickup and delivery of FedEx shipments, package tracking, and conformance to policies and procedures for a wide variety of domestic and international services.

Dan Ștefănescu: Dr. Dan Ștefănescu is a Postdoctoral Research Fellow in the Department of Computer Science of the University of Memphis and the Institute for Intelligent Systems (IIS). As a member of DeepTutor team, his main research activity is dialogue-based Intelligent Tutoring Systems. Previously, Dr. Ștefănescu was a Senior Researcher at the Research Institute for Artificial Intelligence (RACAI) in Bucharest, Romania. He graduated from the Computer Science Faculty of ”A.I. Cuza” University of Iași in 2002 and obtained his MSc in Computational Linguistics from the same university in 2004. In 2010 he was awarded the PhD title (Magna Cum Laude) at the Romanian Academy for a thesis on Knowledge Extraction from Multilingual Corpora. He authored more than 50 papers in peer-reviewed journals and conference proceedings and successfully participated in various software competitions like the Question-Answering competitions organized by Conference and Labs of the Evaluation Forum (CLEF), Microsoft Imagine Cup and Microsoft Speller Challenge. His research work covers various Natural Language Processing topics like: Question Answering, Information Extraction, Word Sense Disambiguation, Connotation/Sentiment Analysis, Collocations/Terminology Identification, Machine Translation, or Query Alteration for Search Engines.

The SCHOLAR Legacy: A New Look at the Affordances of Semantic Networks for Conversational Agents in Intelligent Tutoring Systems

Donald M. Morrison and Vasile Rus

Institute for Intelligent Systems, The University of Memphis, Memphis, Tennessee
{dmmrrson and vrus}@memphis.edu

Abstract. The time is ripe for a new look at the affordances of semantic networks as backbone structures for knowledge representation in intelligent tutoring systems (ITSs). While the semantic space approach has undeniable value, and will likely continue to be an essential part of solutions to the problem of computer-based dialogue with humans, technical advances such the automatic extraction of ontologies from text corpora, now encourage a vision in which intelligent tutoring agents have access to forms of knowledge representation that allow them to more fully "understand" something of what they are talking about with learners. These developments have important implications for key ITS components including the structure of expert domain models, learner models, instructional modules, and dialogue strategies, particularly in respect to issues of transportability across systems. As such, they in turn have important implications for the design of a general-purpose framework such as the U.S. Army's Generalized Intelligent Framework for Tutoring (GIFT).

Keywords: Intelligent tutoring, semantic networks, semantic spaces, ontology extraction.

1 Introduction

The idea that a computer might be programmed to carry on an intelligent conversation with a human emerged in the early days of artificial intelligence, possibly as early as the 1940s, but was articulated most fully in computer pioneer Alan Turing's famous "Turing test" [40] in which a human is invited to carry on a typed conversation with both a hidden human and a machine, and has to decide which is which. A computer program that passes the Turing test is considered to be intelligent. Early programs that were claimed to have passed the test included ELIZA [43], which employed the ping-pong conversational strategies of a Rogerian psychotherapist, thus, allowing ELIZA to be "free to assume the pose of knowing almost nothing of the real world" [p. 42], and PARRY, which was designed to mimic the behavior of a paranoid schizophrenic, and reportedly fooled about half the psychologists who interacted with it [11].

SHRDLU [44] was similar to ELIZA and PARRY in that it was capable of engaging humans in something like natural language dialog, but distinctly different in one important respect; it actually "understood" something about what it was talking about, albeit within a highly restricted domain, a virtual world consisting of a set of wooden blocks of different shapes and colors. SHRDLU could be instructed to pick up a block, put it on top of another block, and would remember which block was now on top.

SCHOLAR [9, 10, 17] is thought to have been the first program to demonstrate the use of mixed-initiative, natural-language conversation for the specific purpose of instruction, and as such is widely considered the first intelligent tutoring system [e.g., 29,14]. The program was intended to demonstrate what Carbonell, a computer scientist at Bolt, Beranek and Newman (BBN), termed an "information-structure-oriented" (ISO) tutorial system. In contrast to "ad hoc-frame oriented" (AFO) systems of the time, which were organized around "frames" of specific text strings, questions, and answers written by a human author, SCHOLAR drew its questions and answers *directly from its own knowledge base*, which, in the examples referenced here, consisted of a set of simple facts about the geography of South America.[8] The following transcript, reported by Freedman [17], gives some sense of how these interactions went:

SCHOLAR: The capital of Chile is Santiago. Correct or incorrect?
STUDENT: Correct.
SCHOLAR: That's right. Approximately what is the area of Brazil?
STUDENT: 2,500,000 square miles.
SCHOLAR: Wrong. Please indicate if the following statement is correct or incorrect:
The area of Paraguay is approx. 47432 square miles.

While it seems unlikely that SCHOLAR fooled anyone, the program was arguably ahead of its time in one important respect. SCHOLAR's knowledge of the world was stored in a *semantic network*, a data structure that featured in the work of Carbonell's colleagues at BBN, Ross Quillian and Allan Collins [32, 12, 13]. Semantic networks do not, in themselves, provide easy solutions to the problem of machine understanding of human language; however, for reasons explained below, there is good reason to take a second look at the various affordances they may offer to designers of general-purpose intelligent tutoring systems (ITSs), including general-purpose frameworks such as GIFT.

2 Affordances of Semantic Networks for Intelligent Tutoring Systems

Researchers in artificial intelligence have explored a range of solutions to the problem of representation of conceptual knowledge, from symbolic representations to purely statistical ones [25,19]. Semantic networks of the type employed by SCHOLAR, where concepts and their relationships are represented as nodes and edg-

[8] Carbonell was born in Uruguay. A second database was developed to provide tutoring for an online text editing system.

es, are arguably closest to symbolic natural language in that noun-predicate-object clusters (semantic triples) are incorporated and preserved. In "semantic space" models, on the other hand, relationships among concepts are represented mathematically. Methods include Latent Semantic Analysis (LSA) [24], Hyperspace Analogue to Language (HAL) [26], Latent Dirichlet Allocation (LDA) [5], Non-Latent Similarity (NLS) [8]; Word Association Space (WAS) [39], and Pointwise Mutual Information (PMI) [33].

In general terms, these semantic space models identify the meaning of a word through "the company it keeps" [15:11], that is, by examining the co-occurrence of words across large numbers of documents and using this data to calculate statistical measures of semantic similarity. This approach has been used successfully in a variety of applications where measures of document similarity are useful, such as in text retrieval and automatic scoring of student essays [25]. In intelligent tutoring applications, probabilistic semantic space engines allow for the automatic creation of domain models as "bags of words" [20]. For example, AutoTutor employs LSA measures of text similarity to evaluate the extent to which a learner's answers to its questions correspond to scripted correct answers consisting of unordered sets of expected words and phrases [42].

When applied to the problem of knowledge representation in intelligent learning systems, the selection of one approach over another results in important trade-offs. Although the choice of probabilistic semantic models in intelligent tutoring systems avoids the time-consuming tasks involved in creating more granular, linguistically encoded models of domain knowledge, it also imposes significant constraints on the functionality of the system, including limits on its ability to engage in true dialog with a human learner, which in turn constrains both its ability to represent what is in the learner's head *and* the nature and quality of the apparent (virtual) social relationship between the agent and the learner.

Most importantly, an agent that relies exclusively on a probabilistic semantic model cannot generate substantive questions of its own, nor can it respond to a learner's questions. Rather, because its knowledge is enclosed in a "black box" [1] it is limited to asking scripted questions with scripted answers, then evaluating the extent to which the learner's answers conform. As a result, it naturally assumes the role of a traditional pedagogue, a teacher who looks only for correct answers to questions.

2.1 Some Recent Developments

In spite of these limitations, in recent years the use of probabilistic, black box semantic models has been favored over semantic network representations, owing, as noted above, largely to the difficulties inherent in laborious manual authoring of useful domain models based on semantic networks [35]. However, over the past decade or so this situation has begun to change in important ways. While the extraction of propositions (semantic triples) from connected text—the building blocks of semantic network solutions—remains as one of the hardest problems in artificial intelligence and machine learning [35,19], considerable progress has been made [e.g., 2, 31, 30, 6, 4].

For example, Berland & Charniak [2] developed an algorithm which, given a seed word such as *car*, and a large corpus of text to mine, identified the following as possible fillers for the slot ___ *is-part-of* ____[*car*]: *headlight*, *windshield*, *ignition*, *shifter*, *dashboard*, *radiator*, *brake*, *tailpipe*, etc. Similarly, Pantel & Ravichandran [31] describe an algorithm for automatically discovering semantic classes in large databases, labeling them, then relating instances to classes in the form X *is-a* Y. For example, for the instances *Olympia Snowe*, *Susan Collins*, and *James Jeffords*, the algorithm settled on *republican*, *senator*, *chairman*, *supporter*, and *conservative* as possible labels, meaning that it could form the basis for assertions such "*Olympia Snowe is a republican*."

Other relevant work includes the corpus of annotated propositional representations in PropBank [30], and AutoProp [6] a tool that has been designed to "propositionalize" texts that have already been reduced to clauses. More recently, members of the DBpedia project [4] have been working to extract semantic triples from Wikipedia itself. As of September 2011, the DBpedia dataset described more than 3.64 million "things," with consistent ontologies for some 416,000 persons, 526,000 places, 106,000 music albums, 60,000 films, 17,500 video games, 169,000 organizations, 183,000 species and 5,400 diseases. A similar project, Freebase, allows users to edit ontologies extracted from Wikipedia [27], while YAGO2 [21] is a knowledge base of similar size (nearly 10 million entities and events, as well as 80 million facts representing general world knowledge) that includes the dimensions of space and time in its ontologies. All of these projects employ a form of semantic network to represent conceptual knowledge.

Given the labor required in building formal representations of procedural knowledge by hand, it is natural to consider the possibility of automatic extraction of production rules from text corpora, using machine learning (data mining) methods similar to those for extracting declarative knowledge. As it turns out, work on this problem is already producing promising results. For example, Schumacher, Minor, Walter, & Bergmann [36] have compared two methods of extracting formal "workflow representations" of cooking recipes from the Web, finding that the frame-based SUNDANCE system [34] gives superior results, as rated by human experts. Song et al. [37] have tested a method for extracting procedural knowledge from PubMed abstracts. Jung, Ryu, Kim, & Myaeng [23] describe an approach to automatically constructing what they call "situation ontologies" by mining sets of how-to instructions from the large-scale web resources eHow (www.eHow.com) and wikiHow (www.wikihow.com).

While the implications of this work for the development of intelligent learning systems remain unclear, the possibilities inherent in semantic data mining of both declarative and procedural knowledge clearly deserve attention. It seems the most likely scenario is that future systems will employ different knowledge representations for different purposes. For example, Rus [35] describes the use of a hybrid solution, Latent Semantic Logic Form (LS-LF), for use in the extraction of expert knowledge bases from corpora such as textbooks. Also, while the use of semantic networks in particular domains may allow an agent to engage in something approaching intelligent conversation regarding these domains, the agent may still need a way of coping with user utterances that it cannot handle in any other way, much as humans make educated, intuitive guesses about the meaning of ambiguous or confusing utterances. For example, Hu & Martindale [22] discuss the use of a semantic vector model as a means of evaluating the relevance and novelty of a given utterance in a series of discourse moves, which is clearly useful in the event that an agent has no other way of evaluating a user's utterance.

2.2 Implications for General-purpose Tutoring Systems

The field of intelligent tutoring has come a long way in the four decades that separate us from the time of SCHOLAR. A recent estimate [28], identified some 370 ITS "architecture families," or which 12 were considered "major architectures," defined as those with at least ten scholarly papers published between the years 2009-2012. However, in spite of these efforts (representing investments of untold millions of taxpayer dollars), the field has not yet had much of an impact on educational practice. The study cited above, for example, estimated less than 1 million users worldwide. To put this in perspective, a recent estimate puts the number of school-age children in the U.S. at 70 million, and in the world at over 1 billion [7].

Important barriers to more widespread adoption and impact of ITSs include two important and related problems. One is the high cost of authoring domain-specific systems, recently estimated to require between 24 and 220 hours of development time for one hour of instruction, with a mean of around 100 hours [16]. A second problem is that ITSs tend to be constructed as "unique, one-of-a-kind, largely domain-dependent solutions focused on a single pedagogical strategy" [38]. Among other things, because components are not shareable, this means that returns on investment in particular systems is limited to whatever impact those particular systems might on their own, like stones tossed into a pond that make no ripples.

The use of semantic networks to represent expert domain knowledge might go far to reduce authoring costs and could also lead to portable expert models, and, by extension, learner models. As we have seen, a considerable amount of work is already going on in the semi-automatic (i.e., supervised) extraction of domain ontologies from text corpora. What this means, conceptually, is that the ontology of a particular domain becomes not just a single person (or team's) unique description of the domain of interest, but a structure that emerges from the way the domain is represented linguistically in some very large number of texts, written by different authors. While it is true that supervised extraction introduces and reflected the biases of the human supervisors, ontologies constructed in this way arguably have much more in common than those constructed entirely from scratch for specific purposes. The ability to extract domain models directly from text corpora also, of course, speeds the development process, and, to the extent that expert models constructed in this way are architecture-independent, they are more likely to acquire general currency than dedicated models developed for the particular purposes of specific systems. Finally, to the extent that learner models, or at least some portion of them, are seen as overlays of expert models (i.e., flawed or incomplete versions of expert maps), these may also become transportable across systems, and because these models can be expressed mathematically, as graphs, it becomes possible to estimate differences between learner models and expert models computationally.

3 Conclusion

While the specific affordances of semantic networks in respect to problems of knowledge representation, learner modeling, and conversational fluency of intelligent agents have yet to be fully explored, and while such structures do not by any means solve fundamental problems, the future is indeed promising. As argued here, the movement to structure the vast store of human knowledge on the Web in the form of explicit ontologies, as evidenced in the Semantic Web project and its many associated technologies, is well underway, and has undeniable momentum. The future of human knowledge representation almost certainly lies in this direction, with some obvious potential benefits to ITS developers. For example, to the extent that expert domain models are conceived as populated ontologies, then it becomes easier to conceive of portable domain models, and, to the extent that a learner models are *also* conceived of as populated ontologies, then learner models can also be portable across systems.

Interestingly, the underpinnings of the Semantic Web originated in the work of Ross Quillian, the same work that SCHOLAR, the ancestor of modern ITSs, was based on. Now that the technology is beginning to catch up with that initial vision, the time has arguably come to take another look at the affordances of semantic networks. In particular, the designers of systems such as GIFT, which seek to provide a general-purpose framework for development of ITS systems of the future, are advised to look carefully at the specific implications of the reemergence and increasing importance of semantic networks as general-purpose structures for representing the knowledge of both experts and learners, and as the basis for bringing these structures into alignment through natural processes of teaching and learning.

References

1. Anderson, J.R. The expert model. In Polson, M. C., & Richardson, J. J. (eds.) Foundations of intelligent tutoring systems. Lawrence Erlbaum. 21-53 (1988)
2. Berland, M., Charniak, E. Finding parts in very large corpora. In Annual Meeting-Association For Computational Linguistics 37, 57-64 (1999, June)
3. Bickmore, T., Schulman, D., Yin, L. Engagement vs. deceit: Virtual humans with human autobiographies. In Intelligent Virtual Agents, pp. 6-19. Springer Berlin/Heidelberg (2009)
4. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S. DBpedia-A crystallization point for the Web of Data. Web Semantics: Science, Services and Agents on the World Wide Web, 7(3), 154-165 (2009)
5. Blei, D. M., Ng, A. Y., Jordan, M. I. Latent dirichlet allocation. The Journal of Machine Learning Research, 3, 993-1022 (2003)
6. Briner, S.W., McCarthy, P.M., McNamara, D.S. Automating text propositionalization: An assessment of AutoProp. In R. Sun & N. Miyake (eds.), Proceedings of the 28th Annual Conference of the Cognitive Science Society, pp. 2449. Austin, TX: Cognitive Science Society (2006)
7. Bruneforth, M. & Wallet, P. Out-of-school adolescents. UNESCO Institute for Statistics. (2010).

8. Cai, Z., McNamara, D. S., Louwerse, M., Hu, X., Rowe, M., Graesser, A. C. NLS: A non-latent similarity algorithm. In Proc. 26th Ann. Meeting of the Cognitive Science Soc., CogSci'04, pp. 180-185 (2004)
9. Carbonell, J. R. AI in CAI: Artificial intelligence approach to computer assisted instruction. IEEE Transactions on Man-Machine Systems 11(4): 190-202 (1970)
10. Carbonell, J. R., Collins, A. M. Natural semantics in artificial intelligence. In Proceedings of the Third International Joint Conference on Artificial Intelligence. IJCAI 73, 344 -351 (1973, August)
11. Colby, K. M., Hilf, F. D., Weber, S., Kraemer, H. C. Turing-like indistinguishability tests for the validation of a computer simulation of paranoid processes. Artificial Intelligence, 3, 199-221 (1972)
12. Collins, A. M., Loftus, E. F. A spreading-activation theory of semantic processing. Psychological review, 82(6), 407 (1975)
13. Collins, A. M., Quillian, M. R. Retrieval time from semantic memory. Journal of verbal learning and verbal behavior, 8(2), 240-247 (1969)
14. Corbett, A. T., Koedinger, K. R., Anderson, J. R. Intelligent tutoring systems. Handbook of human-computer interaction, 849-874 (1997)
15. Firth, John Rupert. A synopsis of linguistic theory, 1930-1955. (1957).
16. Folsom-Kovarik, J. T., S. Schatz, and D. Nicholson. Return on investment: A practical review of ITS student modeling techniques. M&S Journal, Winter Edition (2011): 22-37.
17. Freedman, R. Degrees of mixed-initiative interaction in an intelligent tutoring system. In Proceedings of the AAAI Spring Symposium on Computational Models for Mixed Initiative Interaction, Palo Alto, CA. Menlo Park, CA: AAAI Press (1997)
18. Graesser, A. C., Lu, S., Jackson, G. T., Mitchell, H. H., Ventura, M., Olney, A., Louwerse, M. M. AutoTutor: A tutor with dialogue in natural language. Behavior Research Methods, Instruments, & Computers, 36(2), 180-192 (2004)
19. Graesser, A. C., McNamara, D. S., Louwerse, M. M. Two methods of automated text analysis. Handbook of Reading Research, 34 (2009)
20. Harris, Z. Distributional structure. Word 10 (2/3): 146–62 (1954)
21. Hoffart, J., Suchanek, F. M., Berberich, K., Lewis-Kelham, E., De Melo, G., Weikum, G. Yago2: exploring and querying world knowledge in time, space, context, and many languages. In Proceedings of the 20th international conference companion on World Wide Web, pp. 229-232. ACM (2011, March)
22. Hu, X., Martindale, T. Enhancing learning with ITS-style interactions between learner and content. Interservice/Industry Training, Simulation & Education 2008, 8218, 1-11 (2008)
23. Jung, Y., Ryu, J., Kim, K. M., Myaeng, S. H. Automatic construction of a large-scale situation ontology by mining how-to instructions from the web. Web Semantics: Science, Services and Agents on the World Wide Web, 8(2), 110-124 (2010)
24. Landauer, T. K., Dumais, S. T. A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. Psychological Review , 104 , 211-140 (1997)
25. Landauer, T. K., Laham, D., Foltz, P. W. Automated scoring and annotation of essays with the Intelligent Essay Assessor. Automated essay scoring: A cross-disciplinary perspective, 87-112 (2003)
26. Lund, K., & Burgess, C. Producing high-dimensional semantic spaces from lexical co-occurrence. Behavior Research Methods, Instruments, & Computers, 28(2), 203-208 (1996)
27. Markoff, J. Start-up aims for database to automate web searching. The New York Times (2007-03-09) Retrieved 4/21/2013

28. Nye, B.. Two sigma or two percent: A mapping study on barriers to ITS adoption. (in preparation)
29. Nwana, H. S. Intelligent tutoring systems: an overview. Artificial Intelligence Review, 4(4), 251-277 (1990)
30. Palmer, M., Kingsbury, P., Gildea, D. The Proposition Bank: An annotated corpus of semantic roles. Computational Linguistics, 31, 71-106 (2005)
31. Pantel, P., & Ravichandran, D. Automatically labeling semantic classes. In Proceedings of HLT/NAACL (4), 321-328 (2004, May)
32. Quillian, M.R. Semantic memory. In M. Minsky, (E.), Semantic Information Processing. MIT Press, Cambridge, MA (1968).
33. Recchia, G., Jones, M. N. More data trumps smarter algorithms: Comparing pointwise mutual information with latent semantic analysis. Behavior Research Methods, 41(3), 647-656 (2009)
34. Riloff, E., Phillips, W. An introduction to the sundance and autoslog systems. Technical Report UUCS-04-015, School of Computing, University of Utah (2004)
35. Rus, V. What next in knowledge representation? Proceedings of the International Conference on Knowledge Engineering, Principles and Techniques. Cluj-Napoca (Romania), July 2-4, pp. 1-9 (2009)
36. Schumacher, P., Minor, M., Walter, K., Bergmann, R. Extraction of procedural knowledge from the web: A comparison of two workflow extraction approaches. In Proceedings of the 21st international conference companion on World Wide Web, pp. 739-747. ACM (2012, April)
37. Song, S. K., Choi, Y. S., Oh, H. S., Myaeng, S. H., Choi, S. P., Chun, H. W., and Sung, W. K. Feasibility study for procedural knowledge extraction in biomedical documents. Information Retrieval Technology, 519-528 (2011)
38. Sottilare, R. A., Goldberg, B. S., Brawner, K. W., & Holden, H. K. A Modular Framework to Support the Authoring and Assessment of Adaptive Computer-Based Tutoring Systems (CBTS). In *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)* (Vol. 2012, No. 1). National Training Systems Association. (2012, January).
39. Steyvers, M., Griffiths, T. L., Dennis, S. Probabilistic inference in human semantic memory. Trends in Cognitive Sciences, 10(7), 327-334 (2006)
40. Turing, A. M. Computing machinery and intelligence. Mind, 59(236), 433-460 (1950)
41. VanLehn, K. The behavior of tutoring systems. International journal of artificial intelligence in education, 16(3), 227-265 (2006)
42. Wiemer-Hastings, Peter, Arthur C. Graesser, and Derek Harter. The foundations and architecture of AutoTutor. In Intelligent Tutoring Systems, pp. 334-343. Springer Berlin Heidelberg, 1998.
43. Weizenbaum, J. ELIZA—A computer program for the study of natural language communication between man and machine. Communications of the ACM, 9(1), 36-45 (1966)
44. Winograd, T. Procedures as a representation for data in a computer program for understanding natural language. MIT AI Technical Report 235 (February 1971)

Authors

Dr. Chip Morrison is a Faculty Affiliate at IIS. A graduate of Dartmouth, Dr. Morrison holds an M.A. from the University of Hong Kong and an Ed.D. from Harvard. His current research interests include models of human cognition and learning, and the application of these models to conversation-based intelligent learning systems.

Vasile Rus: Dr. Vasile Rus is an Associate Professor of Computer Science with a joint appointment in the Institute for Intelligent Systems (IIS). Dr. Rus' areas of expertise are computational linguistics, artificial intelligence, software engineering, and computer science in general. His research areas of interest include question answering and asking, dialogue-based intelligent tutoring systems (ITSs), knowledge representation and reasoning, information retrieval, and machine learning. For the past 10 years, Dr. Rus has been heavily involved in various dialogue-based ITS projects including systems that tutor students on science topics (DeepTutor), reading strategies (iSTART), writing strategies (W-Pal), and metacognitive skills (MetaTutor). Currently, Dr. Rus leads the development of the first intelligent tutoring system based on learning progressions, DeepTutor (www.deeptutor.org). He has coedited three books, received several Best Paper Awards, and authored more than 90 publications in top, peer-reviewed international conferences and journals. He is currently Associate Editor of the International Journal on Artificial Intelligence Tools.

XNAgent: Authoring Embodied Conversational Agents for Tutor-User Interfaces

Andrew M. Olney, Patrick Hays, & Whitney L. Cade

Institute for Intelligent Systems & Department of Psychology
365 Innovation Drive
Memphis, Tennessee 38152
{aolney,dphays,wlcade}@memphis.edu
http://iis.memphis.edu

Abstract. Embodied conversational agents are virtual characters that engage users in conversation with appropriate speech, gesture, and facial expression. The high cost of developing embodied conversational agents has led to a recent increase in open source agent platforms. In this paper, we present XNAgent, an open source platform for embodied conversational agents based on the XNA Framework. By leveraging the high-level class structure of the XNA Framework, XNAgent provides a compact implementation that is suitable both as a starting point for the development of a more advanced system and as a teaching tool for AI curricula. In this paper we describe how we created an embodied conversational agent in XNA using skeletal and morph animation, motion capture, and event-driven animation and how this process can facilitate the use of embodied conversational agents in the Generalized Intelligent Framework for Tutoring.

Keywords: XNA, ECA, GIFT, agent, HCI, conversation, interface, tutoring

1 Introduction

It is well known that we unconsciously and automatically interact with computers using social norms [1]. Embodied conversational agents (ECAs) capitalize on this phenomena as characters with human-like communicative capabilities. By doing so, ECAs leverage pointing, gestures, facial expressions, and voice to create a richer human-computer interface. As a result ECAs have been used in diverse AI applications, including education [2], where they form an important part of the tutor-user interface.

ECAs combine research in discourse, computer animation, speech synthesis, and emotion. Consequently ECA systems tend to be costly to build [3] As a result, in the past decade, a great deal of tutoring research has used closed-source platforms such as Microsoft Agent [4], adapted commercial/open source game engines [5], or low-level libraries like OpenGL [6]. These approaches present different types of challenges. Game engines usually have support for basic character animation but lack native lip-sync and fine animation control, and game engines come with a complex API with many features that may not be relevant for education research, e.g. bullet/explosion physics or first-person shooter perspective. Conversely low-level libraries have no similar irrelevant complexity but require designing the AI from the ground up. Given the challenges of both the game-engine and low-level routes, recent researchers have released open source platforms for ECA development [7, 8, 9, 10] based on either game engines or low-level libraries.

The design and development challenges described above for ECAs are manifest in the development of computer-based training environments and have recently been addressed by the Generalized Intelligent Framework for Tutoring Framework [11]. One of the design goals of the Generalized Intelligent Framework for Tutoring (GIFT) is to provide authoring capability for the creation of computer-based training components. One such component is the tutor-user interface, which in modern intelligent tutoring systems often uses an ECA. Accordingly, in this paper we present an open source solution to ECA development that meets the design goals of the GIFT Framework. Rather than use a game engine with its inherent complexities or a low-level library that requires a large investment of initial development, we present an ECA platform that combines the best of these using Microsoft's XNA framework [12]. By providing high-level libraries, a runtime environment for managed code (C#), free development tools, and extensive support in the form of code samples, official forums, and commercially available books at all levels, the XNA framework provides a solid foundation for ECA development. In this the following sections we describe how we implement the face and body of XNAgent using skeletal and morph animation via vertex shaders, motion capture, and event-driven animation. At each step the content creation pipeline is outlined to illustrate how XNAgent may be adapted to new AI contexts. We conclude by considering the design goals of the GIFT Framework and how they are addressed by XNAgent.

2 Face

The face of an ECA can be considered independently of the body in terms of speech, emotions, and facial expressions. The classic reference for facial expression is the Facial Action Coding System, which uses the anatomy of the face, primarily in terms of muscle groups, to define facial action units [13]. While it is certainly possible to create "virtual muscles" and animate with them, a number of other real-time approaches exist which give satisfactory results [14]. Perhaps the most well-known and widely used facial animation approach is morph target animation.

In morph target animation, a version of the head is created for each desired expression. For example, one version for smiling, frowning, or a "w" lip shape. Each of these shapes becomes a target for interpolation, a morph target. If two morph channels exist, e.g. a neutral face and a smiling face, the interpolation between them can be described by the distance between matching vertices across the two faces. In practice, this distance is often normalized as a weight such that a weight of 1 would push the neutral face all the way to happy. The advantage to using morph target animations is that each morph target can be carefully crafted to the correct expression, and then mixtures of morph targets can be used to create huge number of intermediate expressions, e.g. smiling while talking and blinking.

FaceGen Modeler, by Singular Inversions, is a popular software package for creating 3D faces that has been used in psychological research on gaze, facial expression, and attractiveness [15]. FaceGen Modeler contains a statistical model of the human face with approximately one hundred and fifty parameters to vary face shape and texture. Using FaceGen Modeler, a virtual infinite variety of human faces can be created by manipulating these parameters, and for a given custom face FaceGen Modeler can output thirty-nine morph targets including seven emotions and sixteen visemes (the visual correlates of phonemes used for lip-sync). XNAgent uses FaceGen Modeler output, so a correspondingly large variety of faces can be implemented in XNAgent.

Since XNA does not provide native support for morph targets, we have implemented them using vertex shaders. A shader is a program that runs directly on the graphics card. In XNA, shaders are written in High Level Shader Language that resembles the C programming language, and the shaders compile side by side with C#. To implement morph target animation, XNAgent's vertex shaders operate on each vertex on face and perform bilinear interpolation (interpolation on two axes). Thus there are three versions of the XNAgent head loaded at any particular time: a neutral head that was skinned with the body (see Section 3), a viseme head for the current viseme, and an emotion/expression head for the current emotion. It is possible to have more channels for additional morphing, and these are easily added if necessary.

XNAgent utilizes a dynamic, event-driven animation system for facial expressions. Three categories of facial animation are currently supported, including blinking, lip-sync via visemes, and facial expressions. Blinking is implemented using a model of blinking behavior in humans [16] in its own thread. Because the most salient feature of blinking is perhaps that the eyelids cover the eyes, XNAgent imitates blinking through texture animation rather than morph target animation. In texture animation the texture of the face is switched quickly with another version of the face. In the case of blinking the two textures are nearly identical except the blink texture's eyes are colored to match the surrounding skin, thus simulating closed eyes.

Lip-syncing through morph target animation is controlled by the agent's voice, i.e. a text-to-speech synthesizer. Some speech synthesizers generate lip-sync information during synthesis by producing visemes, the visual correlates of phonemes. Each viseme unit typically includes the current viseme and the viseme's duration. In a viseme event handler, XNAgent sets the current viseme morph target and its duration using these values. In the Update() loop, the viseme's time left is decremented by the elapsed time. In the Draw() loop, the viseme morph is expressed with a weight based on the remaining time left. Thus the lip sync remains true independently of the framerate speed of the computer running XNAgent and linearly interpolates between visemes.

Morphing expressions like emotions require a more flexible approach than viseme animations. For example, a smile can be a slow smile that peaks at a medium value, or a rapid smile that peaks at an extreme value. To capture these intuitions, our expression morph animation has parameters for rise, sustain, and decay times, with a maximum weight parameters that specifies what the maximal morph will be during the sustain phase. Currently these three phases are interpolated linearly.

3 Body

Non-facial movements, or gestures, appear to greatly differ from the face greatly in terms of communicative complexity, forming sign language in the extreme case. Our approach is therefore to model the entire body as a collection of joints, such that manipulating the values of these joints will cause the body to move. This common approach to animation is often called skeletal, or skinned animation [17].

In skinned animation a character "shell" is first created that represents a static character. An underlying skeletal structure is created for the shell with appropriate placement of joints and placed inside the shell. The shell and skeleton are then bound together such that a transformation on the underlying skeleton is mirrored in the shell; this result is known as a rigged model. Once a model is rigged, it may be animated by manipulating the skeleton and saving the resulting joint position data. Every saved movement creates a keyframe, and when these keyframes are played back at a rapid rate (e.g. 30 fps) the rigged model will carry out the animated action. Alternatively motion capture technologies can extrapolate joint position data from naturalistic human movement. In this case the resulting animation is still a keyframe animation.

In order to create a body for XNAgent, we used several software packages to form what is commonly known as a 3D authoring pipeline. At each stage of the pipeline there are multiple available techniques and software packages, making navigating this space a complex process. In brief, there are three major phases to creating a body with gestures, namely model creation, rigging, and animation. Model creation can be extremely difficult for non-artists without initial materials to work from. To facilitate the process of body creation, we used the face model generated by FaceGen Modeler together with the FaceGen Exporter to export the face model to the Daz Studio software package. This process seamlessly combines the face and body models and auto-rigs the body with a skeleton. Daz Studio allows for comparable customizations of the body (gender, size, shape) as FaceGen does for the face. In addition, Daz Studio comes with a variety clothing and accessory packs that can be applied to the body in a drag and drop manner. In effect, several hundred hours of 3D authoring can be accomplished by a novice in less than an hour.

In order to create realistic animations, we primarily used the low-cost iPi Desktop Motion Capture system from iPi Soft. The simplest camera configuration for this system uses the Microsoft Kinect camera. Once the motion capture has been recorded by iPi, it can be merged and edited using AutoDesk 3DS Max, where ultimately it is exported for XNA using the kw X-port plugin. A complete description of this process is beyond the space limitations of the current discussion, but a full tutorial, including software installer and step by step slides, is available from the corresponding author's website[9].

In order to achieve similar functionality to interpolating visemes, skinned animation clips require mechanisms for blending and mixing. Simply put, blending is end to end interpolation, like a DJ fading from one song to the next. Mixing breaks the animation into components and plays them simultaneously, like a DJ taking the beat from one song, vocals from another, and playing them together. Blending and mixing can be done simultaneously if clips are playing in different regions of the skeleton

[9] http://andrewmolney.name

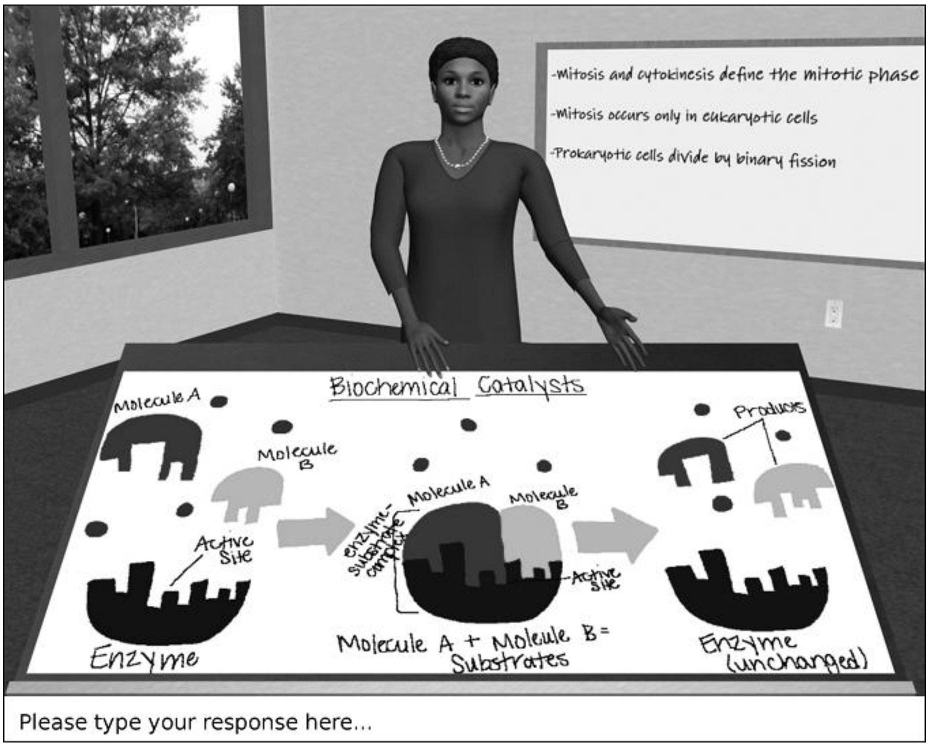
while being blended with other clips in those same regions. XNAgent uses the Communist Animation Library [18] to perform blending and mixing. Currently in XNAgent the skeleton is divided into center, left side, right side, and head regions. These regions are used to represent the following tracks: idle, both arms, left arm, right arm, and head. Animations are assigned to tracks at design time and then played with weights according to what other animations are currently playing in their track. For example, the idle animation consists of motion capture of a person standing and slightly swaying. Some degree of the idle animation is always playing in all the tracks, but when other animations are played in those tracks they are played with a higher weight. Thus lower priority animations like idle will be superseded by higher priority animations in a relatively simple manner.

Animations are triggered in XNAgent by inserting animation tags into the text to speak, either dynamically or manually. When the TTS encounters the tag, it schedules the animation immediately. The mixing properties of the animation are specified in the tag to create new versions of animations, similar to morphing. For example, since the idle animation is always playing, it can be given more weight relative to an arm gesture to create a "beat" gesture [19]. Thus a normal full arm extension animation can be dampened arbitrarily using weighting, bringing the arm closer to the body with increasing weight. In addition, the speed of the animation clip can be modulated to control for the appropriate speed of the beat gesture, since beat gestures are often quick and fluid.

Although XNA has some level of built in support for skinned animations, combining skinned animations with morph target animations requires a custom vertex shader. In XNAgent there are two vertex shaders that operate separately on the head and body of the agent. The head shader applies morphing to calculate a new vertex position and then applies the transformation defined by skinning. This allows the head to be applying morph targets (e.g. speaking) while also nodding or shaking. The second vertex shader focuses strictly on the body and so does not require morphing.

4 Working with XNAgent

One of the most important aspects of any ECA is its ability to integrate into an AI application. Game engines typically don't support integration well and rather present a fullscreen interface for the game, as does XNA. Although text input and other user interface functions can be carried out inside XNA, they are difficult because XNA doesn't provide the native support commonly expected by GUI designers. For example, key presses in XNA are interpreted based on the framerate of the game, meaning that a normal keystroke will produce a double or triple production of letters or numbers. To address the integration issue, XNAgent provides an XNA environment inside a Windows form control. That means that adding XNAgent to an interface is as simple as selecting the XNAgent control from the Visual Studio toolbox and dropping it on a form. The primary method to call on the control is Speak(), which processes both text to speech and animation tags as described in previous sections. In summary, the process for using XNAgent is (1) create a 3D model using the authoring pipeline described above (2) import the model to XNAgent (3) call XNAgent from your application using the Speak() method. We have previously integrated XNAgent into the Guru intelligent tutoring system shown in Figure 1 and conducted a number of experiments [20].



-Mitosis and cytokinesis define the mitotic phase

-Mitosis occurs only in eukaryotic cells

-Prokaryotic cells divide by binary fission

Please type your response here...

Figure 1: XNAgent running in the Guru intelligent tutoring system.

We argue that XNAgent fulfills many if not all of the design goals for GIFT authoring components [11]. XNAgent decreases the effort of authoring ECAs through its 3D authoring pipeline. Similarly it decreases the skills required for authoring ECAs by making authoring a drag-and-drop process, rather than a pixel-by-pixel process. XNAgent's animation framework allows authors to organize their knowledge about pedagogical animations and helps structure pedagogical animations. Perhaps most importantly in a research environment, XNAgent supports rapid prototyping of ECAs with different properties (gender, size, or clothing) for different pedagogical roles (teacher, mentor, or peer). XNAgent supports standards for easy integration with other software as a Windows form control. By cleanly separating domain-independent code from specific 3D model and animation content, XNAgent promotes reuse. Finally XNAgent leverages open source solutions. Not only is XNAgent open source, but every element in its 3D authoring pipeline either has a freeware version or is free for academic use. Moreover, the recent version of MonoGame, an open source implementation of XNA, promises to make XNAgent cross platform to desktop and mobile devices beyond the Windows desktop.

5 Conclusion

In this paper we have described the XNAgent platform for developing embodied conversational agents. Unlike existing ECA platforms that require either low level graphics programming or the use of complex game engines, XNAgent is written using a high level framework (XNA). Our contribution to this research area is in showing how to implement appropriate speech, gesture, and facial expression using skeletal and morph animation via vertex shaders, motion capture, and event-driven animation. We argue that the XNAgent platform fulfills most of the authoring design goals for GIFT with respect to authoring ECAs. It is our hope that XNAgent will be used by adopters of GIFT to facilitate creation of dialogue based tutoring systems that use ECAs.

6 Acknowledgments

The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A080594 and by the National Science Foundation, through Grant BCS-0826825, to the University of Memphis. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education or the National Science Foundation.

7 References

1. Nass, C., Steuer, J., Tauber, E.R.: Computers are social actors. In: Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence. CHI '94, New York, NY, ACM (1994) 72–78
2. Dunsworth, Q., Atkinson, R.K.: Fostering multimedia learning of science: Exploring the role of an animated agent's image. Computers & Education 49(3) (November 2007) 677–690
3. Heloir, A., Kipp, M.: Real-time animation of interactive agents: Specification and realization. Applied Artificial Intelligence 24 (July 2010) 510–529
4. Graesser, A.C., Lu, S., Jackson, G.T., Mitchell, H., Ventura, M., Olney, A., Louwerse, M.M.: AutoTutor: A tutor with dialogue in natural language. Behavioral Research Methods, Instruments, and Computers 36 (2004) 180–193
5. Rowe, J.P., Mott, B.W., W. McQuiggan, S.W., Robison, J.L., Lee, S., Lester, J.C.: CRYSTAL ISLAND: A Narrative-Centered learning environment for eighth grade microbiology. In: Workshops Proceedings Volume 3: Intelligent Educational Games, Brighton, UK (July 2009) 11–20
6. Lester, J.C., Voerman, J.L., Towns, S.G., Callaway, C.B.: Deictic believability: Coordinating gesture, locomotion, and speech in lifelike pedagogical agents. Applied Artificial Intelligence 13 (1999) 383–414
7. Damian, I., Endrass, B., Bee, N., André, E.: A software framework for individualized agent behavior. In: Proceedings of the 10th international conference on Intelligent virtual agents. IVA'11, Berlin, Heidelberg, Springer-Verlag (2011) 437–438

8. Heloir, A., Kipp, M.: Embr — a realtime animation engine for interactive embodied agents. In: Proceedings of the 9th International Conference on Intelligent Virtual Agents. IVA '09, Berlin, Heidelberg, Springer-Verlag (2009) 393–404
9. de Rosis, F., Pelachaud, C., Poggi, I., Carofiglio, V., De Carolis, B.: From Greta's mind to her face: modelling the dynamics of affective states in a conversational embodied agent. International Journal of Human-Computer Studies 59(1-2) (2003) 81–118
10. Thiebaux, M., Marsella, S., Marshall, A.N., Kallmann, M.: SmartBody: behavior realization for embodied conversational agents. In: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 1, Estoril, Portugal, International Foundation for Autonomous Agents and Multiagent Systems (2008) 151–158
11. Sottilare, R.A., Brawner, K.W., Goldberg, B.S., Holden, H.K.: The generalized intelligent framework for tutoring (GIFT). Technical report, U.S. Army Research Laboratory â" Human Research & Engineering Directorate (ARL-HRED) (October 2012)
12. Cawood, S., McGee, P.: Microsoft XNA game studio creator's guide. McGraw-Hill Prof Med/Tech (2009)
13. Ekman, P., Rosenberg, E.: What the face reveals: basic and applied studies of spontaneous expression using the facial action coding system FACS. Series in affective science. Oxford University Press (1997)
14. Noh, J., Neumann, U.: A survey of facial modeling and animation techniques. Technical Report 99-705, University of Southern California (1998)
15. N'Diaye, K., Sander, D., Vuilleumier, P.: Self-relevance processing in the human amygdala: gaze direction, facial expression, and emotion intensity. Emotion 9(6) (December 2009) 798–806
16. Pelachaud, C., Badler, N., Steedman, M.: Generating facial expressions for speech. Cognitive Science 20 (1996) 1–46
17. Gregory, J.: Game engine architecture. A K Peters Series. A K Peters (2009)
18. Alexandru-Cristian, P.: Communist animation library for xna 4.0 (December 2010)
19. McNeill, D.: Hand and mind: what gestures reveal about thought. University of Chicago Press (1992)
20. Olney, A., D'Mello, S., Person, N., Cade, W., Hays, P., Williams, C., Lehman, B., Graesser, A.: Guru: A computer tutor that models expert human tutors. In Cerri, S., Clancey, W., Papadourakis, G., Panourgia, K., eds.: Intelligent Tutoring Systems. Volume 7315 of Lecture Notes in Computer Science., Springer Berlin / Heidelberg (2012) 256–261

Authors

Andrew Olney is presently an assistant professor in the Department of Psychology at the University of Memphis and the associate director of the Institute for Intelligent Systems. His primary research interests are in natural language interfaces. Specific interests include vector space models, dialogue systems, grammar induction, robotics, and intelligent tutoring systems.

Patrick Hays is a research assistant at the University of Memphis. He is a recent graduate with a BA in Psychology. Patrick's work focuses on 3D animation, 3D modeling, graphic arts, and human-computer interaction.

Whitney Cade is a graduate student in the Department of Psychology at the University of Memphis. Her research interests include intelligent tutoring systems, expert tutors, pedagogical strategies, 3D agents, and machine learning.

AIED 2013 Workshops Proceedings
Volume 8

Formative Feedback in Interactive Learning Environments (FFILE)

Workshop Co-Chairs:

Ilya Goldin
Carnegie Mellon University, USA

Taylor Martin
Utah State University, USA

Ryan Baker
Teachers College Columbia University, USA

Vincent Aleven
Carnegie Mellon University, USA

Tiffany Barnes
North Carolina State University, USA

https://sites.google.com/site/ffileworkshop/

Preface

Educators and researchers have long recognized the importance of formative feedback for learning. Formative feedback helps learners understand where they are in a learning process, what the goal is, and how to reach that goal. While experimental and observational research has illuminated many aspects of feedback, modern interactive learning environments provide new tools to understand feedback and its relation to various learning outcomes.

Specifically, as learners use tutoring systems, educational games, simulations, and other interactive learning environments, these systems store extensive data that record the learner's usage traces. The data can be modeled, mined and analyzed to address questions including when is feedback effective, what kinds of feedback are effective, and whether there are individual differences in seeking and using feedback. Such an empirical approach can be valuable on its own, and it may be especially powerful when combined with theory, experimentation or design-based research. The findings create an opportunity to improve feedback in educational technologies and to advance the learning sciences.

The FFILE workshop aims to advance and encourage research on using data to understand and improve feedback and interactive learning environments. The organizers hope to facilitate the exchange of ideas and the growth of the community of researchers who are interested in these topics. As evidenced by the publications in this volume, using data to understand and improve feedback is important and timely. The papers cover a variety of topics, including rubric-based automated assessment of student drawings of chemical reactions (Rafferty et al.), IRT-based modeling of the effect of feedback on analogical reasoning in children (Stevenson et al.), and an assessment technique for student responses that relies on student participation (Jordan et al.).

Each submission to the workshop was reviewed by three members of a Program Committee, which included the co-chairs and representatives of academia, industry and independent research institutions. The co-chairs thank the Program Committee for diligent reviewing and service.

The co-chairs also thank Erin Walker and Chee-Kit Looi, the AIED 2013 Tutorial and Workshop Chairs, and Andrew Olney and Phil Pavlik, the AIED 2013 Local Arrangements Chairs, for their tireless assistance in helping us organize the workshop.

The workshop will include talks, posters, demos, and interactive activities. The organizers hope that the workshop will be of interest to the wider AIED community.

June, 2013
Ilya Goldin, Taylor Martin, Ryan Baker, Vincent Aleven, Tiffany Barnes

Program Committee

Co-Chair: Ilya Goldin, *Carnegie Mellon University, USA* (goldin@cmu.edu)
Co-Chair: Taylor Martin, *Utah State University, USA* (taylormartin@usu.edu)
Co-Chair: Ryan Baker, *Teachers College Columbia University, USA* (baker2@exchange.tc.columbia.edu)
Co-Chair: Vincent Aleven, *Carnegie Mellon University, USA* (aleven@cs.cmu.edu)
Co-Chair: Tiffany Barnes, *North Carolina State University, USA* (tmbarnes@ncsu.edu)

William Cope, *University of Illinois, USA*
Albert Corbett, *Carnegie Mellon University, USA*
Davide Fossati, *Carnegie Mellon University in Qatar, Qatar*
Neil Heffernan, *Worcester Polytechnic Institute, USA*
Pamela Jordan, *University of Pittsburgh, USA*
Sandra Katz, *University of Pittsburgh, USA*
Michael D. Kickmeier-Rust, *Graz University of Technology, Austria*
Young-Jin Lee, *University of Kansas, USA*
Chas Murray, *Carnegie Learning, USA*
Susanne Narciss, *Technische Universitaet Dresden, Germany*
Niels Pinkwart, *Clausthal University of Technology, Germany*
Steve Ritter, *Carnegie Learning, USA*
Valerie Shute, *Florida State University, USA*
John Stamper, *Carnegie Mellon University, USA*
Denise Whitelock, *The Open University, UK*
Caroline Wylie, *Educational Testing Service, USA*

Table of Contents

Full Papers

Posters and Demos

Automating Guidance for Students' Chemistry Drawings

Anna N. Rafferty
Computer Science Division
University of California
Berkeley, CA 94720
rafferty@cs.berkeley.edu

Libby Gerard
Graduate School of Education
University of California
Berkeley, CA 94720
libby.gerard@gmail.com

Kevin McElhaney
Graduate School of Education
University of California
Berkeley, CA 94720
kevin777@berkeley.edu

Marcia C. Linn
Graduate School of Education
University of California
Berkeley, CA 94720
mclinn@berkeley.edu

ABSTRACT

Generative educational assessments such as essays or drawings allow students to express their ideas. They provide more insight into student knowledge than most multiple-choice items. Formative guidance on generative items can help students engage deeply with material by encouraging students to effectively revise their work. Generative items promote scientific inquiry by eliciting a variety of responses and allowing for multiple correct answers, but they can be difficult to automatically evaluate. We explore how to design and deliver automated formative guidance on generative items requiring precollege students to draw the arrangement of atoms before and after a chemical reaction. The automated guidance is based on a rubric that captures increasing complexity in student ideas. Findings suggest that the automated guidance is as effective at promoting learning as teacher-generated guidance, measured both by immediate improvement on the revised item and pre- to post-test improvement on a near-transfer item. Immediate and delayed delivery of automated guidance are equally effective for promoting learning. These studies demonstrate that embedding automated guidance for chemistry drawings in online curricula can help students refine their understanding. Providing automated guidance can also reduce the time teachers spend evaluating student work, creating more time for facilitating inquiry or attending to the needs of individual students.

Keywords

formative feedback | automatic assessment | chemistry education

1. INTRODUCTION

One of the promises of computer assisted education is the ability to provide timely guidance to students that is adapted to their particular mistakes. Such adaptive formative feedback is provided by human tutors [18], and has been shown to be an important principle in designing computerized tutors [1, 2]. This guidance can scaffold student understanding and address common errors that lead different students to express the same incorrect response. While the majority of computerized tutors provide formative feedback in some form [11, 26], this guidance is often limited to selection tasks or numeric answers. These kinds of answers are easy to evaluate yet may encourage students to recall facts rather than distinguish and integrate ideas.

Generative tasks, in contrast, elicit students' range of ideas and encourage them to use evidence to sort out ideas in order to create a coherent explanation. Mintzes, Wandersee, and Novak point to the fact that generative assessments can provide a fuller picture of students' conceptual understanding and drive students towards "making meaning" rather than memorizing facts [19]. Generative tasks are difficult to evaluate due to the variety of responses and possibilities for multiple ways to express the correct answer. Evaluating student work is time consuming and requires content expertise. Subsequently it is often not possible for teachers to provide detailed guidance to all students [5].

In this paper, we explore how automated formative guidance on student-generated drawings can improve students' conceptual understanding of chemical reactions. By constraining students to use virtual atom stamps, rather than drawing the atoms themselves, we limited the degree to which student drawings could vary while still allowing for expression of different conceptual views. We designed an algorithm to automatically evaluate students' conceptual views, and provided targeted guidance to improve understanding.

We begin by reviewing some of the relevant literature on formative feedback as well as the theoretical framework, knowledge integration, in which our work is grounded. We then describe the drawing tasks that students completed as part of an inquiry-based activity concerning global climate change and the highly accurate automated scoring system we developed. We demonstrate how the automated guidance affects student learning through two classroom studies: one explores the effect of automated guidance compared to teacher-generated guidance, and the other investigates whether immediate or delayed automated guidance is more effective.

2. BACKGROUND

There has been a great deal of work on the design and use of formative feedback. We briefly overview some of the most relevant literature on formative feedback for science learning, as well as the knowledge integration framework, which is the pedagogical theory underlying the design of our assessment and guidance.

2.1 Formative Feedback

Formative assessment can help teachers to recognize students' level of understanding and adapt instruction. Ruiz-Primo and Furtak [21] found that teachers' informal use of this type of assessment was related to their students' performance on embedded assessment activities, suggesting that this monitoring can indeed help teachers boost student learning. Guidance based on these assessments provides a way to help students to improve their understanding and recognize gaps or inconsistencies in their ideas [10].

While formative assessment and guidance can be helpful for learning, it is difficult to determine how to design this guidance for generative and open-ended tasks. These tasks facilitate a variety of student responses, and the best form of guidance for promoting learning and conceptual understanding based on students' current knowledge is unclear. Some work has had success at automatically scoring student-generated short answers (e.g., [3],[13]), leading to the potential for conceptual guidance based on these scores. In the science domain, automated feedback has also been effective at driving student learning when creating and revising concept maps [24]. For inquiry learning, there has been significant interest in how to effectively scaffold student learning using technology [20]. While often not aimed directly at guidance, machine learning techniques have been employed to automatically recognize effective inquiry learning skills [22]. Our work adds to this body of literature on formative feedback in open-ended science tasks by demonstrating that drawing tasks in which students pictorially represent scientific ideas are amenable to automatic evaluation. We test how different ways of providing guidance affect student learning.

2.2 Knowledge Integration

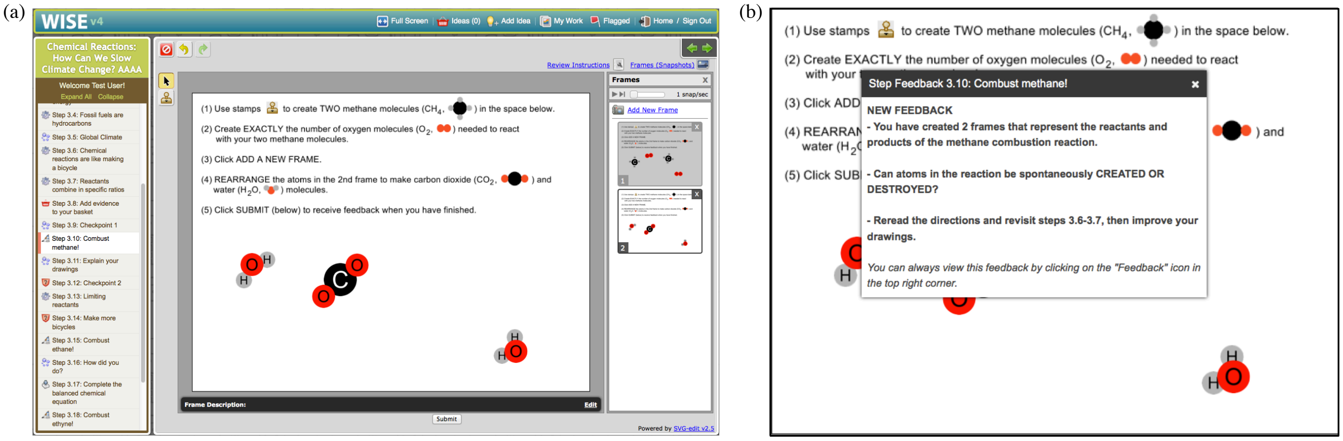
The drawing tasks we examine are part of a chemical reactions unit [7] built in the Web-based Science Inquiry Environment (WISE) [16]. This environment is based on the theory of knowledge integration [15]. Knowledge integration is based on constructivist ideas that focus on building on students' prior knowledge and helping them to connect new concepts with this knowledge, even if some of this prior knowledge is non-normative (e.g.,[27]). Knowledge integration consists of four main processes: eliciting existing student ideas, adding new ideas, distinguishing ideas, and sorting ideas into cohesive understandings [14]. Within WISE, these processes are targeted by activities within an inquiry-based learning module. Each module is organized around a central topic, such as understanding climate change, and the activities may include answering multiple choice or short answer questions, watching a visualization, or creating a drawing to illustrate a scientific phenomenon. For instance, the chemical reactions unit contains visualizations of how energy from the sun is reflected by the Earth and transformed into heat energy. This visualization may add to students' existing ideas as well as help them to see cases that are not accounted for by these existing ideas. Later in the unit, students' understanding is challenged through the introduction of new concepts, such as pollution, into both the visualization and the general investigation of why climate change occurs. This adds new ideas to the student's existing model and prompts revision of the student's ideas to form a more complete understanding. The knowledge integration framework has been the building block for a number of WISE units, and has also been revised and used for pedagogical design in other settings [8, 25].

In the context of knowledge integration, generative tasks elicit students' existing ideas and help them to clarify and distinguish their ideas from one another. Through this process, they may form more cohesive conceptual understandings. For example, a student might make a drawing or write a textual explanation of the visualization she observed. This prompts her to pull out individual ideas and consider how to connect what she saw in the visualization with her prior knowledge. Formative guidance can assist students by prompting them to revise their ideas and evaluate their consistency with normative scientific ideas, which may be articulated or referred to in the feedback [17]. When this guidance is based on students' own ideas, as articulated in their initial response to the activity, it can directly help students to develop criteria for distinguishing between normative and non-normative ideas and push students to integrate ideas rather than holding separate, conflicting conceptions [16].

3. DRAWING CHEMICAL REACTIONS

We focus our investigation of formative feedback on students' drawings of chemical reactions. These drawings show students' particulate understanding of how atoms are rearranged in a reaction. Past work has shown that learning multiple models of chemical reactions and providing students with ways of visualizing the particles involved in the reactions can help to strengthen student understanding [9, 23]. The drawing tasks are part of a WISE unit entitled *Chemical Reactions: How Can We Help Slow Climate Change?*, which focuses on students' understanding of chemical reactions [7]. As shown in Figure 1(a), these drawing tasks ask students to draw the arrangement of atoms before and after a chemical reaction; one of the tasks focuses on the combustion of methane while the other involves the combustion of ethane. The WISE Draw screen provides students with "stamps" for each atom; for instance, the methane reaction problem includes stamps for oxygen, carbon, and hydrogen. Students must choose how many of each atom to add to their drawing and arrange the atoms to reflect how they are grouped into molecules. Students then create a new frame in their drawing to show the products of the reaction. The drawings enable students to articulate their ideas about chemical reactions and to work with a different model of these reactions than the typical equation based format.

Both the methane and ethane tasks ask the student to show the combustion of oxygen and a hydrocarbon, resulting in the products carbon dioxide and water. In the methane drawing, students are asked to draw two methane molecules


(a)
WISE v4
Full Screen
Ideas (0)
Add Idea
My Work
Flagged
Home / Sign Out
Chemical Reactions: How Can We Slow Climate Change? AAAA
Welcome Test User!
Expand All
Collapse
Step 3.4: Fossil fuels are hydrocarbons
Step 3.5: Global Climate
Step 3.6: Chemical reactions are like making a bicycle
Step 3.7: Reactants combine in specific ratios
Step 3.8: Add evidence to your basket
Step 3.9: Checkpoint 1
Step 3.10: Combust methane!
Step 3.11: Explain your drawings
Step 3.12: Checkpoint 2
Step 3.13: Limiting reactants
Step 3.14: Make more bicycles
Step 3.15: Combust ethane!
Step 3.16: How did you do?
Step 3.17: Complete the balanced chemical equation
Step 3.18: Combust ethyne!
Review Instructions
Frames (Snapshots)
Frames
1 snap/sec
Add New Frame
1
2
O
H
C
Frame Description:
Edit
Submit
Powered by SVG-edit v2.5
(b)

(1) Use stamps to create TWO methane molecules (CH_4,) in the space below.

(2) Create EXACTLY the number of oxygen molecules (O_2,) needed to react with your two methane molecules.

(3) Click ADD A NEW FRAME.

(4) REARRANGE the atoms in the 2nd frame to make carbon dioxide (CO_2,) and water (H_2O,) molecules.

(5) Click SUBMIT (below) to receive feedback when you have finished.

Step Feedback 3.10: Combust methane! ×

NEW FEEDBACK

- You have created 2 frames that represent the reactants and products of the methane combustion reaction.

- Can atoms in the reaction be spontaneously CREATED OR DESTROYED?

- Reread the directions and revisit steps 3.6-3.7, then improve your drawings.

You can always view this feedback by clicking on the "Feedback" icon in the top right corner.

Figure 1: The WISE drawing environment. (a) A screenshot of a student drawing. Students place atom stamps on the central canvas to show the molecules at the beginning and end of a chemical reaction. On the right side of the screen, the two frames that the student has created are shown. (b) The student drawing canvas with automated guidance. The student has submitted her drawing, and a pop up box appears with adaptive textual feedback to help her develop her conceptual understanding of chemical reactions.

and as many oxygen molecules as are required for complete combustion of the methane. This item thus requires students to reason about how many oxygen molecules each methane molecule reacts with. For the ethane drawing, students are told to illustrate ten oxygen molecules and two ethane molecules as the reactants, and then to rearrange them to form the products. This leaves three oxygen molecules that are unchanged by the reaction.

4. PROVIDING GUIDANCE ON STUDENT DRAWINGS

Since the drawing tasks assess important conceptual ideas about chemical reactions and students frequently make errors on these tasks, they are a natural target for providing students with formative feedback. Our goal is to provide conceptual guidance that targets errors that the student has made. This requires detecting errors in the drawing and creating guidance for each category of conceptual errors.

4.1 Evaluating Student Drawings

To evaluate student drawings, we created an algorithm that processes each drawing and assigns it a score. We used a development set of 98 drawings from past students, half from each item, to determine the most common errors and to tune the parameters of the scoring algorithm. Of these 98 drawings, 45% were correct, as marked by a human evaluator.

Examination of the student drawings showed many similar errors across students. We grouped these errors into conceptual categories, shown in Table 1. Category 0 includes drawings that do not have two frames, one for the reactants and one for the products. In some cases, this may be due to difficulties using the drawing interface. Category 1 corresponds to lack of conservation of mass. Student drawings with this error have different atoms in the reactant and product frames. Category 2 corresponds to drawings that conserve mass, but have incorrect reactants. This may be due to having the wrong number of molecules, or to having atoms incorrectly arranged into molecules. Category 3 refers to drawings that have correct reactants, but incorrect products. For instance, a student might combust only one methane molecule, incorrectly leaving one methane and two oxygen molecules in the products. Category 4 includes drawings that are nearly correct, but where molecules are overlapping; for example, four oxygen atoms might be arranged in a square, rather than arranged in two distinct groups. Finally, Category 5 includes correct drawings.

In order to facilitate feedback across a variety of chemical reaction drawings, we separated the scorer into a scoring algorithm and a specification file. The scoring algorithm maps the drawing into one of the six categories described above, drawing information from the specification file to determine the correct configuration of atoms into molecules and what molecules are correct for each frame. In the methane case, for example, the specification file lists four allowed molecules: oxygen, methane, carbon dioxide, and water. Each molecule is defined by the atoms that it includes and how these atoms touch one another. For instance, the specification file indicates that carbon dioxide includes one carbon and two oxygen atoms, and each oxygen atom must touch the carbon atom. The specification file also lists the correct reactants and products for the given reaction. While this level of expressivity was sufficient for our tasks, which have a single correct set of molecules that should be present in each frame, the specification file and scorer could easily be extended to specify non-unique correct answers, such as requiring that the products should have twice as many of one molecule as another.

Student drawings are saved as SVG strings, an XML-based vector image format, which facilitates automatic processing. Each string indicates how many frames exist, what stamps are in each frame, and the location of each stamp. The specification file lists how stamps (image files) correspond to atoms, so the string effectively indicates the location of each atom in the drawing. The automated scoring algorithm has three stages: pre-processing, identifying molecule groupings,

Criteria	**0**	**1**	**2**	**3**	**4**	**5**
Two frames		✓	✓	✓	✓	✓
Conserves atoms			✓	✓	✓	✓
Correct reactants				✓	✓	✓
Correct products					✓	✓
Groupings clear						✓
Rate in dev. set	11%	19%	16%	5%	3%	45%

Table 1: The scoring rubric. Each level adds an additional criterion that must be met. The bottom row indicates the proportion of drawings in the development set with each score.

and assigning a numerical score. Pre-processing removes stamps that are outside of the viewable image area, often due to a student dragging a stamp offscreen rather than deleting it. This stage also removes duplicate stamps that have identical or almost identical center locations; this can occur when a student double-clicks to place a stamp. The pre-processing steps thus makes the SVG string correspond more closely to the image as a viewer would perceive it.

After pre-processing, atom stamps are grouped into molecules, and the frames are annotated with the atoms and molecules that they contain. Atoms are part of the same molecule if they are visually grouped. This is indicated by the atoms directly touching, with atoms in one molecule not touching atoms in another molecule. Small spaces between the atoms in a molecule and small amounts of overlap are ignored by our algorithm due to our focus on conceptual errors; these issues are more likely to be due to the constraints of the medium than evidence of student misunderstanding.

Algorithmically, the grouping of atoms into molecules is computed via depth-first search and by solving a constraint satisfaction problem [28]. Depth-first search computes the connected components of the drawing, where a component is connected if all images in that component are within ϵ of at least one other image in the component; given small $\epsilon > 0$, atoms can be in the same molecule but not directly touch. Components are then matched to molecules, where a match is valid if the identity of the atoms in the specification and in the drawing are the same and if the touching relations given in the problem specification are satisfied; this is implemented as constraint satisfaction. If one connected component can only be recognized as consisting of several molecules, the drawing is marked as having overlapping molecules unless the overlap is less than some constant. Again, this constant allows us to ignore small amounts of overlap.

Based on the annotations of the molecules and atoms in each frame, the numerical score for the drawing is computed based on the rubric in Table 1. For instance, if the number of atoms of each type changes between the first and second frames, the drawing is given a score of 1. If the drawing conserves mass but reactants are not correct, the drawing is given a score of 2, regardless of whether the products are correct. A score of 4 is given only if all atoms in the frames are correct, and the scorer recognized that the correct molecules were present but overlapping.

We evaluated the accuracy of the algorithm on both the development set and on pilot data from 251 student drawings. In both cases, the drawings were scored by a trained human scorer, and these scores were compared to the automated scores. On the development set, the automated score matched the human score on 97% of the drawings. Accuracy was very similar for the pilot data, which was not used in the creation of the scorer: automated scores matched the human score on 96% of the drawings.

4.2 Creating Guidance from Scores

Given that the scoring algorithm is quite accurate, we can provide guidance based on the conceptual understanding that the student has displayed in the drawing. For each of the six possible scores, we designed a textual feedback message to help students revise their drawing. We chose to use textual feedback to facilitate a comparison between automated and teacher-generated guidance. The WISE platform supports teacher guidance by allowing teachers to view student work and type comments to each student group.

The textual feedback was designed to promote knowledge integration by recognizing students' normative ideas and helping them to refine and revise their non-normative ideas [16]. Drawings that were scored as having some conceptual error (scores 0-4) all received textual feedback of a similar format. First, a correct feature of the drawing was recognized, anchoring the guidance with students' prior knowledge. For example, a student who received a score of 2 would be praised for conserving mass, since this is the conceptual feature that bumped the student from a score of 1 to 2. The textual feedback then posed a question targeting the student's conceptual difficulty, such as identifying what molecules should be present in the reactant frame; this elicits student ideas about the topic of difficulty. Finally, the feedback directed students to a relevant step earlier in the unit, and encouraged them to review the material in that step and then to revise their drawing. This promotes adding new ideas and distinguishing normative and non-normative ideas. The feedback for a score of 1 is shown in Figure 1(b).

5. STUDY 1: EFFECTIVENESS OF AUTOMATED GUIDANCE

To test the effectiveness of our automated guidance system, we compared student learning when given automated or teacher-generated guidance. In this study, automated guidance was provided to students upon request, taking advantage of the fact that automation facilitates immediate feedback. Based on evaluation of the existing student drawings, we believed the automated scorer would have relatively high accuracy, but the guidance it can provide is still less specific than that which teachers can provide. The teachers could adjust guidance for individual students, while there were only six different automated feedback messages that a student might receive. Since prior work has had mixed results concerning whether specific or general feedback is more helpful(e.g., [6],[12]), it is not clear whether the lack of specificity in the automated guidance will be a disadvantage.

5.1 Methods

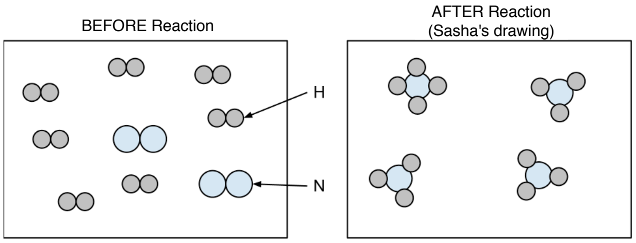
5.1.1 Participants

A total of 263 students used the WISE unit and completed both the pre- and post-tests.

Two N_2 molecules and seven H_2 molecules in a CLOSED container react according to the balanced equation:

$$N_2 + 3H_2 \rightarrow 2NH_3$$

The box on the left shows the container BEFORE the reaction. The box on the right shows Sasha's drawing of the container AFTER the reaction.



Give as many reasons as you can why Sasha's drawing is INCORRECT.

Figure 2: Item from the pre- and post-test related to drawing chemical reactions. Students are asked to examine Sasha's drawing and explain why the drawing is incorrect. The drawing task is similar to those in the unit, but asks students to evaluate rather than generate the drawing and requires integrating the equation and the drawing.

5.1.2 Study design

Students were assigned on a full-class basis to receive either automated or teacher-generated guidance. Two teachers from the same public middle school participated in the study, using the WISE activity in their eighth grade physical science classes. The activity took approximately five hours, spread over multiple class periods. The first teacher had 139 students in five classes; three of these classes received automated guidance and two received teacher guidance. The second teacher had 124 students, also spread over five classes; again, three of the classes were assigned automated guidance and two were assigned teacher guidance. This led to 155 students in the automated condition and 108 students in the teacher guidance condition. Students used WISE in groups of between one and three students; there were 71 groups in the automated condition and 58 in the teacher condition, although a small number of students in these groups did not complete the pre-test or the post-test.

All students experienced the same activities in the WISE unit except for the draw steps. On the two draw steps, all students received the same instructions, except that students in the automated condition were told to click the "Submit" button when they wished to receive feedback. When students clicked this button, they were warned that they only had two chances to receive feedback and to confirm that they wanted to proceed. After confirming, a pop-up box with the textual feedback appeared, as in Figure 1(b). Students could close the feedback or re-open it to view their existing feedback at any time.

Students in the teacher-generated guidance condition did submit their work. Instead, teachers provided feedback to these students using the WISE Grading Tool after the students made a drawing. When students signed in to the activity the following day, they were informed that they had received feedback, and teachers also reminded the students to revise their drawings based on the comments. This condition was intended to mirror how teachers usually give feedback to student work in WISE. Due to time constraints, students in this condition received only one round of feedback.

Students in all conditions completed a pre- and post-test assessment. Both assessments contained the same items. As shown in Figure 2, one of these items asked students to examine a drawing of a chemical reaction and to explain why the drawing was incorrect. This item addresses some of the same conceptual skills as the drawing tasks in the unit, and thus can be used as a transfer measure of student learning from the draw activities. Unlike the WISE unit, these assessments were completed by students individually.

5.2 Results

Overall, students improved their drawings by 0.9 points after receiving guidance, as computed via the automated algorithm. An analysis of variance of student scores on the drawing items with factors for revision that received feedback versus final revision and feedback condition, as well as a random factor for student group, showed that there was a main effect of revision ($F(1, 142) = 68.8$, $p < .001$), indicating the improvement was significant. However, there was not a main effect of condition: improvement was nearly identical for students who received automated guidance and those who received teacher guidance, and both groups had similar initial scores.

While amount of improvement on the drawing items is similar for both conditions, one might be concerned that students in the automated guidance condition have an advantage on this metric since their feedback is directly based on the scoring rubric. Comparison of the proportion of groups revised an incorrect drawing to be correct suggests that this is unlikely to be the case: 27% of groups who were initially incorrect revised their drawing to be correct in the automated condition, compared to 30% in the teacher-feedback condition. Thus, comparable number of students were able to completely correct their work in both conditions.

The improvement from pre- to post-test of student answers on the item concerning evaluation of another student's drawing provides another way of comparing student learning across conditions (see Figure 2). Student answers on this item were evaluated using the rubric in Table 2. This rubric gives higher scores to student answers that include more correct ideas and that connect conceptual ideas with features from the drawing, consistent with the knowledge integration focus on creating a cohesive conceptual understanding. While some of these concepts, such as conservation of mass, were addressed in the drawing items in the unit, the item asks students to go beyond the initial drawing tasks by articulating the connections between the drawing and the equation for the chemical reaction. Students in both conditions improved significantly on this item from pre- to post-test: an average of 0.37 points for students in the automated condition ($t(154) = 4.63$, $p < .005$) and an average of 0.27 points for students in the teacher-feedback condition ($t(107) = 2.93$, $p < .01$). An analysis of variance showed that there was no main effect of feedback type on amount of improvement. Like the results of the improvement in drawings, this suggests that the automated feedback is as helpful for student learning as the teacher-generated feedback.

Inspection of the teacher comments revealed that one teacher gave substantially more detailed and conceptually focused comments than the other. This teacher used a relatively

Score	Criteria
1	Blank or no scientific ideas.
2	Invalid scientific ideas or only correct ideas about products, failing to explain why the products are incorrect.
3	Incomplete scientific ideas: isolated ideas about too few hydrogen in Sasha's drawing or about product identity, without connecting to concepts.
4	One complete statement linking a feature of Sasha's drawing with why it is incorrect.
5	Identification of at least two errors, with complete statements linking the features of Sasha's drawing with why they are incorrect.

Table 2: The knowledge integration scoring rubric for the pre- and post-test item.

small number of comments for all students, customizing these comments slightly on a case by case base, and each one tended to focus on a particular conceptual issue. For example, one comment was *"You have only made one frame to represent the products and reactants. Your first frame should be for the reactants. A second frame should be made for the products. Follow the directions on the top of the page."* This comment combines procedural elements connecting to the student drawing with conceptual ideas. In contrast, the second teacher tended to give short comments that were solely procedural or solely conceptual. These comments commonly directed students to read the directions or stated a concept in isolation, such as the comment *'Conservation of mass?"*. These comments may have been too terse to help students connect concepts with their drawings.

Due to these differences in comments, we analyzed how effective the feedback was at helping students based on what type of feedback they received as well as which teacher they had in the teacher-feedback condition. An analysis of variance on the amount of improvement in drawing scores from initial feedback to final revision, with a factor for feedback type (automated, Teacher 1, or Teacher 2) and a random factor for student group, showed that feedback condition did have an effect on amount of improvement ($F(2, 127) = 4.4$, $p < .05$). As shown in Figure 3, students who received more cohesive guidance (Teacher 1) improved more than students in the other conditions, and students who received automated guidance improved more than students who received terse guidance (Teacher 2). Note that this is not an overall difference between response to guidance based on whether students were in a class with Teacher 1 versus Teacher 2: students in the automated condition showed similar improvement across teachers. While this interaction was not significant for the pre- to post-test improvement, the same trend held: students who received feedback from Teacher 1 improved an average of 0.37 points, students in the automated condition improved 0.35 points, and students who received feedback from Teacher 2 improved 0.12 points.

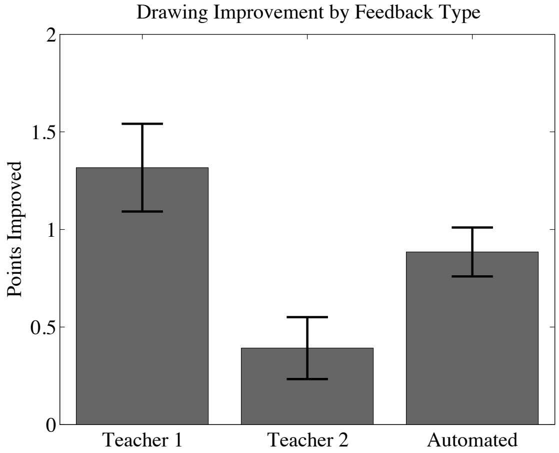


Figure 3: Improvement on drawing scores based on type of feedback received. Error bars indicate one standard error.

6. STUDY 2: TIMING OF GUIDANCE

The previous study showed that automated guidance is comparable to teacher-generated guidance in helping students to revise their drawing and improving post-test scores. However, the two types of guidance were not administered under the same timing schedule: automated guidance was given to students when they asked for it, while teacher guidance was given to students at a fixed delay. We hypothesized that immediate guidance would be more engaging and motivating to students, but delayed guidance might boost retention by allowing students to space their studying of the concepts. Students who are frustrated with the problem may also benefit from a chance to do other activities before receiving guidance. To explore these issues, we conducted a new study in which all students received automated guidance, but some were given the guidance immediately, just as in the automated condition in Study 1, while others received the guidance at a delay, following the same pattern as the teacher guidance in Study 1.

6.1 Methods

6.1.1 Participants

A total of 88 students used the WISE unit and completed both the pre- and post-tests.

6.1.2 Study design

Students were assigned to the immediate or delayed guidance conditions on a full-class basis. All classes were taught by the same teacher in a public high school. He used the activity in his four ninth grade basic chemistry classes. Two classes were assigned to the immediate guidance condition, and two were assigned to the delayed guidance condition. As in Study 1, students completed the activity in groups of one to three students; there were 30 groups in the immediate condition and 27 groups in the delayed condition.

The immediate guidance condition in this study was identical to the automated condition in Study 1. The delayed guidance was provided to students after they had completed their initial drawings, and was added to the grading tool overnight. When students signed into the activity the following day, they were informed that they had new feedback and shown the textual comments. In both cases, the comments students received were based on the score of their drawing, and the text was identical to that of Study 1. Students in the immediate guidance condition could submit their drawing up to two times; due to time constraints, students in the delayed condition received only a single round of feedback.

The pre- and post-test had the same items as in Study 1 and were again completed by students individually.

6.2 Results

Students showed similar improvements in their drawings across conditions. Students in the immediate condition improved their drawing scores by an average of 0.65 points, while students in the delayed condition improved their drawing scores by an average of 0.81 points. A repeated-measures analysis of variance including factors for revision (initial versus final) and guidance condition showed that there was a main effect of revision ($F(1, 65) = 25.2$, $p < .001$), but no significant effect of condition.

In Study 1, we collapsed across the two drawing items as students showed similar improvements across items. However, in this study, there was a trend towards greater improvement on the ethane item for students in the delayed condition versus the immediate condition, while both types of guidance resulted in similar improvement on the methane item. A repeated measures analysis of variance on the amount of improvement with factors for guidance condition and item showed that the interaction between the two factors was marginally significant ($F(1, 52) = 3.44$, $p = .0695$). One reason for this interaction may simply be the placement of these items in the unit: ethane occurs after methane, late in Activity 3 of the WISE unit. Students in the immediate condition may be rushing through the ethane item in order to finish, while students in the delayed condition come back to the items on a later day. Yet, other factors could also contribute to this difference, such as frustration in low-performing students due to the repeated interactive sequences or some item-specific factor.

On the post-test item asking students to evaluate Sasha's drawing, students showed small improvements from their pre-test scores, with an average improvement of 0.19 points. A repeated measures analysis of variance with factors for pre- versus post-test and feedback condition showed that both main effects were significant (pre- versus post test: $F(1, 86) = 4.58$, $p < .05$; condition: $F(1, 86) = 4.12$, $p < .05$). Closer examination revealed relatively little improvement for students in the delayed condition (an average of 0.073 points) compared to an improvement of 0.30 points for students in the immediate condition; by chance, students in the delayed condition also began with higher pre-test scores, although their initial drawing scores were similar.

Overall, this study suggests that immediate and delayed guidance have similar effects on student revision, and immediate guidance may be more helpful for retention and transfer based on the pre- to post-test improvement. Given the difference in effectiveness between the two conditions for improvement on the methane and ethane items, we plan to investigate whether changing the placement of the items within the activities reduces the differences between immediate and delayed guidance. More broadly, we will explore whether students might be helped by different guidance timing for some types of drawing items versus others.

7. DISCUSSION

Formative guidance can help students to improve their understanding of a topic and focus their efforts on the material that is most critical given their current knowledge. We investigated how to provide this guidance in the context of constrained drawing tasks. These tasks allow students to articulate their ideas, including misunderstandings, more fully than multiple choice questions, but are harder to evaluate automatically and too time consuming for teachers to evaluate in many classrooms. We found that by constraining the space of feedback to target six levels of conceptual understanding, we could classify the drawings automatically and help students to improve their understanding. We now turn to some possible next steps for providing formative guidance on drawing items using our automated scoring algorithm.

In our initial studies, we focused on textual feedback in order to compare automated and teacher-generated guidance. However, one of the benefits of a computer-based system is the ability to give other types of guidance, such as interactive activities or guidance that combines text and images. These types of guidance might be more engaging for students, and provide more help for those students who are less motivated or struggle to understand the text-based conceptual feedback. We are currently exploring guidance in the form of interactive activities that place students in the role of evaluating a drawing rather than generating it, just as in the post-test assessment item. The specific activity provided is based on the score of the student's initial drawing.

Another area that we would like to explore in future work is whether more specific or detailed guidance might be helpful for some students. We have observed that some students find it challenging to connect the text-based conceptual feedback with their own drawings. While some level of difficulty is desirable in order to push students to make connections and revise their understanding [4], guidance that is incomprehensible to students is unlikely to help them learn. The automated scoring algorithm provides the potential to scaffold students in their attempt to uncover what is wrong. For instance, if the student has incorrectly grouped some atoms, the algorithm could show the student only the relevant portion of the screen and ask them to explain why that portion was incorrect. This would still prompt students to reflect on their drawings and understanding, but would more closely connect the guidance to their own work. Creating connections between the drawings and the chemistry concepts was common in the guidance of the more effective teacher, suggesting that strengthening these connections in the automated guidance would promote student learning.

The issue of timing and agency when giving feedback remains another useful area for exploration. In Study 2, we compared immediate feedback versus delayed feedback for students, where feedback timing was independent of drawing quality. To better understand how timing of guidance affects learning, we hope to conduct experiments in which timing is based on the score of the current drawing or particular characteristics of students' previous work. These customizations may also allow some students to choose when they would like guidance (as in the immediate condition in

Study 2) while automatically providing guidance to others.

Automatically scoring generative tasks in computerized tutors can be difficult, but is usually a prerequisite of providing adaptive formative feedback on the tasks. In this work, we created an automated scorer for a particular type of constrained yet generative drawing task. This scorer is easily customized to evaluate new drawing items that follow the same pattern as those in the unit, and is able to detect common conceptual errors that students make. Drawing on the knowledge integration pattern, we developed textual guidance for these conceptual errors. Our studies show that that this automated guidance results in comparable learning as guidance given by a teacher. The automated scorer facilitates experimentation with different types of formative feedback, allowing us to test hypotheses about what types of guidance are most effective for promoting understanding in open-ended science activities.

Acknowledgements. This research was supported by a DoD NDSEG fellowship to ANR and by NSF grant number DRL-1119670 to MCL.

8. REFERENCES

[1] J. Anderson, C. Boyle, R. Farrell, and B. Reiser. Cognitive principles in the design of computer tutors. *Modelling Cognition*, pages 93–133, 1987.

[2] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier. Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, 4(2):167–207, 1995.

[3] Y. Attali and J. Burstein. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, 4(3), 2006.

[4] R. A. Bjork. Memory and metamemory considerations in the training of human beings. In *Metacognition: Knowing about knowing*, pages 185–205. The MIT Press, Cambridge, MA, 1994.

[5] P. Black and D. William. Assessment and classroom learning. *Assessment in Education*, 5(1):7–74, 1998.

[6] D. L. Butler and P. H. Winne. Feedback and self-regulated learning: A theoretical synthesis. *Review of Educational Research*, 65(3):245–281, 1995.

[7] J. Chiu and M. Linn. Knowledge integration and wise engineering. *Journal of Pre-College Engineering Education Research (J-PEER)*, 1(1):1–14, 2011.

[8] E. A. Davis and J. S. Krajcik. Designing educative curriculum materials to promote teacher learning. *Educational Researcher*, 34(3):3–14, 2005.

[9] A. G. Harrison and D. F. Treagust. Learning about atoms, molecules, and chemical bonds: A case study of multiple-model use in grade 11 chemistry. *Science Education*, 84(3):352–381, 2000.

[10] J. Hattie and H. Timperley. The power of feedback. *Review of educational research*, 77(1):81–112, 2007.

[11] K. Koedinger, J. Anderson, W. Hadley, and M. Mark. Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8(1):30–43, 1997.

[12] K. R. Koedinger and V. Aleven. Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review*, 19(3):239–264, 2007.

[13] C. Leacock and M. Chodorow. C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, 37(4):389–405, 2003.

[14] M. Linn, B. Eylon, and E. Davis. The knowledge integration perspective on learning. *Internet Environments for Science Education*, pages 29–46, 2004.

[15] M. C. Linn. Chapter 15: The knowledge integration perspective on learning and instruction. In *The Cambridge handbook of the learning sciences*, pages 243–264. Cambridge University Press, New York, NY, 2004.

[16] M. C. Linn and B. Eylon. *Science Learning and Instruction: Taking Advantage of Technology to Promote Knowledge Integration.* Routledge, 2011.

[17] M. C. Linn and B. S. Eylon. *Science Education: Integrating Views of Learning and Instruction*, pages 511–544. Lawrence Erlbaum Associates, Mahwah, NJ, 2006.

[18] D. Merrill, B. Reiser, M. Ranney, and J. Trafton. Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *The Journal of the Learning Sciences*, 2(3):277–305, 1992.

[19] J. J. Mintzes, J. H. Wandersee, and J. D. Novak. *Assessing science understanding: A human constructivist view.* Academic Press, 2005.

[20] C. Quintana, B. J. Reiser, E. A. Davis, J. Krajcik, E. Fretz, R. G. Duncan, E. Kyza, D. Edelson, and E. Soloway. A scaffolding design framework for software to support science inquiry. *The Journal of the Learning Sciences*, 13(3):337–386, 2004.

[21] M. A. Ruiz-Primo and E. M. Furtak. Exploring teachers' informal formative assessment practices and students' understanding in the context of scientific inquiry. *Journal of Research in Science Teaching*, 44(1):57–84, 2007.

[22] M. A. Sao Pedro, R. S. de Baker, J. D. Gobert, O. Montalvo, and A. Nakama. Leveraging machine-learned detectors of systematic inquiry behavior to estimate and predict transfer of inquiry skill. *User Modeling and User-Adapted Interaction*, 23(1):1–39, 2013.

[23] P. Schank and R. Kozma. Learning chemistry through the use of a representation-based knowledge building environment. *Journal of Computers in Mathematics and Science Teaching*, 21(3):253–279, 2002.

[24] J. R. Segedy, J. S. Kinnebrew, and G. Biswas. The effect of contextualized conversational feedback in a complex open-ended learning environment. *Educational Technology Research and Development*, 61(1):71–89, 2013.

[25] S. Sisk-Hilton. *Teaching and Learning in Public: Professional Development through Shared Inquiry.* Teachers College Press, 2009.

[26] J. Slotta and M. Linn. *WISE science: Web-based inquiry in the classroom.* Teachers College Press, 2009.

[27] J. P. Smith III, A. A. Disessa, and J. Roschelle. Misconceptions reconceived: A constructivist analysis of knowledge in transition. *The Journal of the Learning Sciences*, 3(2):115–163, 1994.

[28] E. Tsang. *Foundations of Constraint Satisfaction.* Academic Press London, 1993.

Estimating the Effect of Web-Based Homework

Kim Kelly, Neil Heffernan, Cristina Heffernan
Worcester Polytechnic Institute
kkelly@wpi.edu

Susan Goldman, James Pellegrino, Deena Soffer-Goldstein
University of Illinois-Chicago

ABSTRACT

Traditional studies of intelligent tutoring systems have focused on their use in the classroom. Few have explored the advantage of using ITS as a web-based homework (WBH) system, providing correctness-only feedback to students. A second underappreciated aspect of WBH is that teachers can use the data to more efficiently review homework. Universities across the world are employing these WBH systems but there are no known comparisons of this in K12. In this work we randomly assigned 63 thirteen and fourteen year olds to either a traditional homework condition (TH) involving practice without feedback or a WBH condition that added correctness feedback at the end of a problem and the ability to try again. All students used ASSISTments, an ITS, to do their homework but we ablated all of the intelligent tutoring aspects of hints, feedback messages, and mastery learning as appropriate to the two practice conditions. We found that students learned reliably more in the WBH condition with an effect size of 0.56. Additionally, teacher use of the homework data lead to a more robust and systematic review of the homework. While the resulting increase in learning was not significantly different than the TH review, the combination of immediate feedback and teacher use of the data provided by WBH resulted in increased learning compared to traditional homework practices. Future work will further examine modifications to WBH to further improve learning from homework and the role of WBH in formative assessment.

Keywords

Intelligent tutoring system, immediate feedback, homework, effect size, formative assessment

1. INTRODUCTION

Several studies have shown the effectiveness of intelligent tutoring systems when used in the classroom [9 & 11], reporting effect sizes up to 0.78. However, very few studies have explored the effectiveness of ITS when used as homework. Cooper et al. [3] highlight the point that poorly conceived homework does not help learning. Therefore it was very encouraging when Van Lehn et al. [12] presented favorable results when ANDES, an ITS, was used in this fashion. Yet, most systems are not currently designed to be used for nightly homework. Computer aided instruction (CAI), which gives all students the same questions with immediate end-of-question feedback is more applicable than complex ITS for nightly homework as teachers can easily build the content from textbook questions or worksheets. Kulik and Kulik's [5] meta-analysis reviewed CAI and reported an effect size of 0.3 for simple computer based immediate feedback systems. However, these studies were not in the context of homework use and did not focus on how teachers use the data to respond to student performance. Web-based homework systems (WBH) like WebAssign (www.webassign.com) are commonly used in higher ed. These systems are similar to web based computer aided instruction (CAI), providing students immediate feedback and reports to teachers. While VanLehn et al. [12] reported on three such systems used at the higher ed level for physics, there are no studies that we know of at the K12 level that allow this contrast.

Despite the relatively low effect sizes reported in Kulik and Kulik [5], WBH holds promise for improving learning from homework by tailoring practice to individual performance. Doing so enables individuals to get corrective feedback so they can focus on areas where they are not successful. Shute [8] reviews the plethora of studies and theoretical frameworks developed around understanding the role of feedback for student learning. However, teacher use of the feedback was not a focus. Black and William [1] have focused on formative assessments, with an eye on informing the teacher and giving feedback to students. The cognitive science literature suggests that letting students practice the wrong skill repeatedly on their homework is detrimental to learning. In this study we look to measure the effect on learning by comparing simple WBH to a traditional homework (TH) condition representing the type of practice that millions of students perform every night in America and probably around the world. Additionally, we explore how the teacher can use the data to modify and improve instruction.

The current study employed ASSISTments.org, an intelligent tutoring system that is capable of scaffolding questions, mastery learning, and hint and feedback messages [9]. However, for this study, we ablated those features creating an "end-of-problem-correctness-only" feedback system for homework in the WBH condition. The system was also used for the TH condition by further removing the correctness feedback thus emulating traditional paper and pencil homework assignments. ASSISTments is currently used by thousands of middle and high school students for nightly homework. Many teachers enter the textbook homework problems and answers into ASSISTments so their students can receive immediate feedback on the homework and the teachers can then access item reports detailing student performance. This allows for focused classroom review. In the current study we were also interested in examining the effects of teacher review of homework performance based on information derived from the ASSISTments system under each of the two different homework conditions. The goal was to estimate the additional effects of teacher-mediated homework review and feedback following each of the two homework practice conditions – TH and WBH – and also study differences in how teachers might approach homework review given variation in student performance following each type of homework practice.

2. EXPERIMENTAL DESIGN

Participants were 63 seventh grade students, who were currently enrolled in an eighth grade math class, in a suburban middle school in Massachusetts. They completed the activities included in the study as part of their regular math class and homework. Students were assigned to conditions by blocking on prior knowledge. This was done by ranking students based on their overall performance in ASSISTments prior to the start of the study. Matched pairs of students were randomly assigned to either the TH (n=33) or WBH (n=30) condition.

The study began with a pre-test that was administered at the start of class. This pretest and all the rest of the materials for this study are archived via WebCite so others can see the exact materials, videos and anonymous data at tinyurl.com/AIED2013 [4]. This test consisted of five questions, each referring to a specific concept relating to negative exponents. Students were then given instruction on the current topic. That night, all students completed their homework using ASSISTments (see Kelly, 2012 to experience exactly what students did). The assignment was designed with three similar questions in a row or triplets. There were five triplets and five additional challenge questions that were added to maintain ecological validity for a total of twenty questions. Each triplet was morphologically similar to the questions on the pre-test.

Students in the WBH condition were given correctness-only feedback at the end of the problem. Specifically, they were told if their answer was correct or incorrect. See Kelly [4] to see what these materials looked like and to be able to "play student" in either condition. If a student answered a question incorrectly, he/she was given unlimited opportunities to self-correct, or he/she could press the "show me the last hint" button to be given the answer. It is important to emphasize that this button did **not** provide a hint; instead it provided the correct response, which was required to proceed to the next question.

Students in the TH condition completed their homework using ASSISTments but were simply told that their answer was recorded but were not told if it was correct of not (it says "Answer recorded"). It is important to note that students in both conditions saw the exact same questions and both groups had to access a computer outside of school hours. The difference was the feedback received and the ability for students in the WBH condition to try multiple times before requesting the answer.

The following day all students took PostTest1. This test consisted of five questions that were morphologically similar to the pre-test. The purpose of this post-test was to determine the benefit of feedback while doing their homework. At that point, students in the WBH condition left the room and completed an unrelated assignment. To mimic a common homework review practice, students in the TH condition were given the answers to the homework, time to check their work and the opportunity to ask questions. This process was videotaped and can be seen in Kelly (2012). After all of the questions were answered (approximately seven minutes) students in the TH condition left the room to complete the unrelated assignment and students in the WBH condition returned to class. The teacher used the item report, generated by ASSISTments to review the homework. Common wrong answers and obvious misconceptions guided the discussion. This process was videoed and can be seen at Kelly [4]. The next day, all students took PostTest2. This test was very similar to the other assessments as it consisted of five morphologically similar questions. This post-test can be found at Kelly [4]. The purpose of this test was to measure the value-added by the different in-class review methods.

3. RESULTS

Several scores were derived from the data collected by the ASSISTments system. Student's HW Average was calculated based on the number of questions answered correctly on the first attempt divided by the total number of questions on the assignment (20). Partial Credit HW Score accounted for the multiple attempts allowed in the WBH condition. Students were given full credit for answers, provided they did not ask the system for the response. The score was calculated by dividing the number of questions answered without being given the answer by the number of total questions on the homework assignment (20). Time Spent was calculated using the problem log data generated in ASSISTments and is reported in minutes. Times per action are truncated at five minutes. Recall that the homework assignment was constructed using triplets. Learning Gains within the triplets were computed by adding the points earned on the third question in each triplet and subtracting the sum of the points earned on the first question in each triplet.

3.1 Learning Gains From Homework

One student, who was absent for the lesson, was excluded from the analysis (n=63). A t-test comparing the pre-test scores revealed that students were balanced at the start of the study (t(61)=0.29, p=0.78). However, an ANCOVA showed that students in the WBH condition reliably outperformed those in the TH condition on both PostTest1 (F(1,60)=4.14, p=0.046) and PostTest2 (F(1,60)=5.92, p=0.018) when controlling for pre-test score. See Table 1 for means and standard deviations. If the difference was reliable a Hedge corrected effect size was computed using CEM [2]. The effect sizes do not take into account pretest. The key result for posttest2 of 0.56 effect size had a confidence interval of between 0.07 and 1.08.

A comparison of HW Average shows that students scored similarly (F(1,60)=0.004, p=0.95). An ANCOVA reveled that when calculating homework performance using the Partial Credit HW Score, students in the WBH condition performed reliably better than those in the TH condition (F(1,60)=17.58, p<0.0001). This suggests that with unlimited attempts, students are able to self-correct, allowing them to outperform their counterparts. Similarly, comparing Learning Gains revealed that students with correctness feedback and unlimited attempts to self-correct learned reliably more while doing their homework (F(1,60)=45.72, p<0.0001).

Table 1: Means, standard deviations (in parenthesis), and effect size for each measure by condition. *Notes a reliable difference.

	TH	WBH	*p*-value	Effect Size
Pre-Test	9% (17)	7% (14)	0.78	NA
PostTest1	58% (27)	69% (21)	0.046*	0.52
PostTest2	68% (26)	81% (22)	0.018*	0.56
HW Average	61% (20)	60% (15)	0.95	NA
Partial Credit HW Score	61% (20)	81% (18)	0.0001*	1.04
Time Spent (mins)	22.7 (9.6)	23.2(6.2)	0.96	NA
Learning Gains	0.03 (0.9)	1.73(1.1)	0.0001*	2.21

A review of the item report further describes this difference in learning gains. As expected, students in the TH condition continued to repeat the same mistake each time the question was encountered resulting in three consecutive wrong responses. Conversely, students in the WBH condition may have repeated the mistake once or twice but rarely three times in a row, accounting for the learning. While this behavior appears in four out of the five triplets, triplet 1 was analyzed in depth to explain this finding. See Table 2 for an in depth review of Triplet 1 and Figure 1 to see how the teacher observed this finding using the item report.

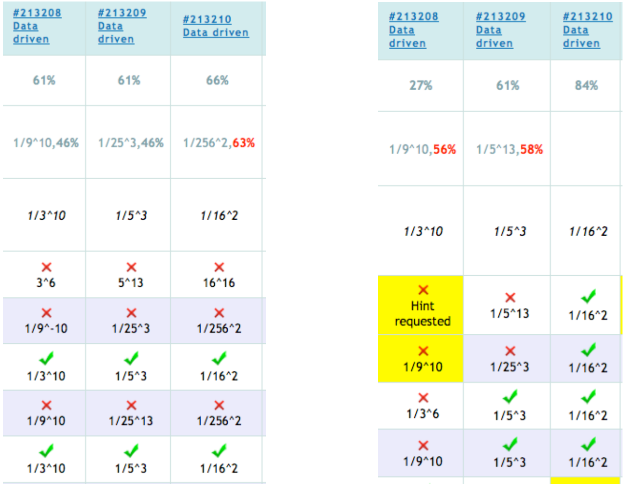
2: An in depth review of Triplet 1.

	WBH	TH
Got the first correct and the last one correct (already knew)	8	17
Got the first one wrong and last one correct (learned)	18	4
Got the first one correct and the last one wrong (unlearned?)	1	2
Got both the first one and the last one wrong (Failed to Learn)	4	9
Total	**31**	**32**

The first thing that we want to point out is that students in the WBH condition had a significantly lower percentage correct on the first item. To demonstrate this finding an in depth review of triplet 1 is provided. Eight of these students requested the answer on the first question in triplet 1. Presumably students in the WBH condition would use the hint button when they were not sure of the answer. However, in the TH condition, there was no such button, therefore perhaps students were more likely to take other steps when they were confused. These steps might have included looking at class notes, asking a parent or calling a friend for help. While there is no data to explain

Additionally, when looking at students in the WBH condition that could demonstrate learning (they got the first one wrong), 18 out of 22 students (80% of students) demonstrated learning. In one sense this learning benefit might be overestimated, as there were some interesting differences in response behavior between the conditions. Specifically, response time for the initial response shows that perhaps students' approach the problems differently. We analyzed the time it took students to type in their first response on question 4, and found that students in the TH condition took longer (121 seconds) than students in the WBH condition (89 seconds). In fact, the TH condition had 34% of students take over two minutes to generate their first response while the WBH condition only had 17% of students take that long. This difference was not statistically significant. We speculate that this is due to the fact that students in this condition knew they would have multiple attempts to correctly answer the question and that there was no penalty for answering incorrectly on the first attempt. This indicates that students in the WBH condition may have a higher percentage of incorrect first responses due to less thorough processing and would account for the higher number of students who seemingly already knew the material in the TH condition.

The ability to attempt each question multiple times is unique to students in the WBH condition. We suggest that this feature may play an important role in the presented learning gains. While this specific feature was not empirically tested in this study, we can only speculate on its effect. However, it is important to note that students in the WBH condition had on average 49 attempts (standard deviation=24) to answer the 20-question homework assignment. The fewest attempts made by any student was 25 and the most was 140. The average number of times the answer was requested was 4 was a standard deviation of 3.5. This suggests that students in the WBH condition took advantage of the ability to try questions multiple times to learn the material without requesting the correct answer.



#213208 Data driven	#213209 Data driven	#213210 Data driven
61%	61%	66%
1/9^10,46%	1/25^3,46%	1/256^2,63%
1/3^10	1/5^3	1/16^2
✗ 3^6	✗ 5^13	✗ 16^16
✗ 1/9^-10	✗ 1/25^3	✗ 1/256^2
✓ 1/3^10	✓ 1/5^3	✓ 1/16^2
✗ 1/9^10	✗ 1/25^13	✗ 1/256^2
✓ 1/3^10	✓ 1/5^3	✓ 1/16^2

#213208 Data driven	#213209 Data driven	#213210 Data driven
27%	61%	84%
1/9^10,56%	1/5^13,58%	
1/3^10	1/5^3	1/16^2
✗ Hint requested	✗ 1/5^13	✓ 1/16^2
✗ 1/9^10	✗ 1/25^3	✓ 1/16^2
✗ 1/3^6	✓ 1/5^3	✓ 1/16^2
✗ 1/9^10	✓ 1/5^3	✓ 1/16^2

Figure 1: The item report for the control condition (on the left) and experimental condition (on the right) for triplet 1, showing the percent of students answering each question correctly, common wrong answers, the correct answer and several rows of student data.

We were not expecting that correctness only feedback was going to be time efficient. But in fact, students in both conditions spent the same amount of time to complete their homework (F(1,60)=0.002, p=0.96). However, it appears that the time spent was apportioned differently in the conditions. Specifically, the TH condition took longer to generate a first response, but the WBH condition took time making multiple attempts as well as requesting the answer. It seems that students in the TH group spend more time thinking about the problem but the WBH group can get the problem wrong, and then use their time to learn the content.

3.2 Learning Gains from Homework Review

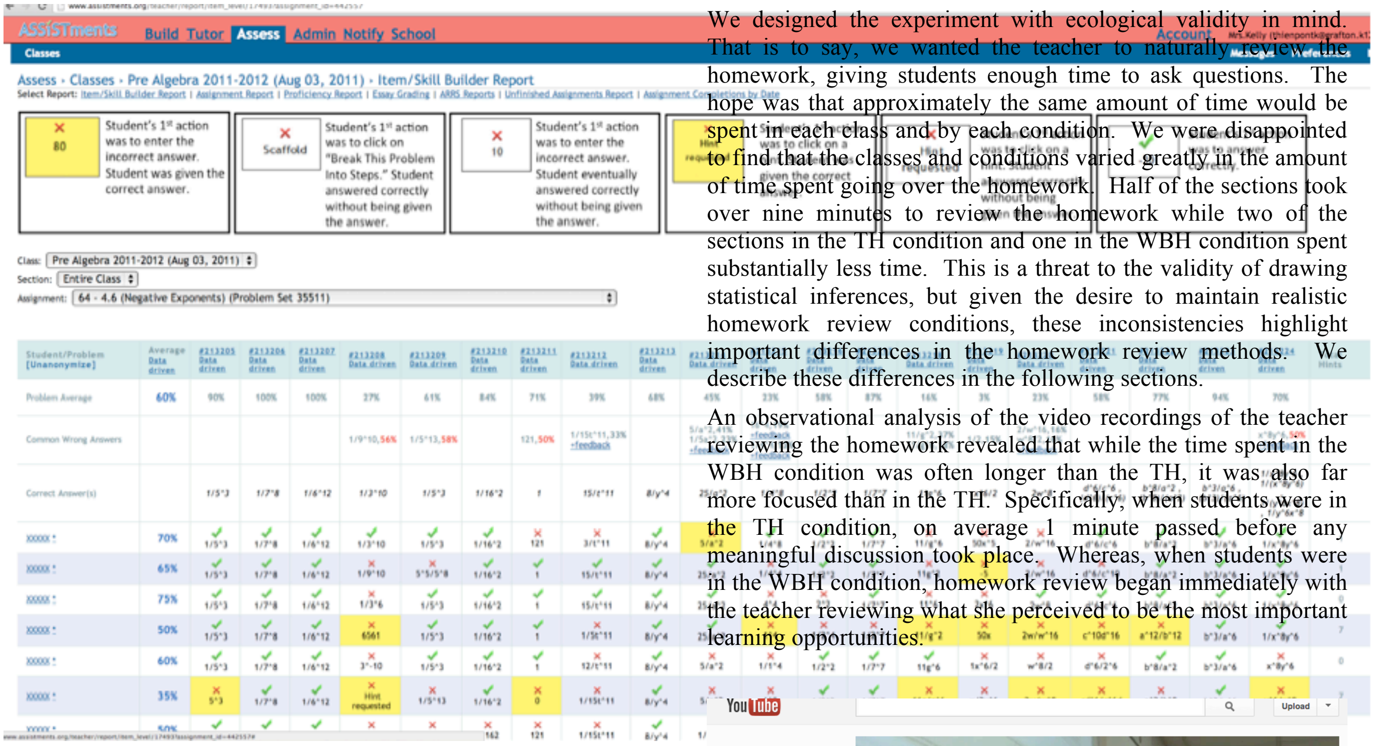
To address the second research question of the effectiveness of using the data to support homework review, a paired t-test revealed that students in both conditions did reliably better on PostTest2 than on PostTest1 (t(62)=3.87, p<0.0001). However, an ANCOVA revealed that when accounting for PostTest1 scores, there is not a reliable difference by condition in the gains from PostTest1 to PostTest2 (F(1,60)=2.18, p=0.15). This suggests that both methods of reviewing the homework lead to substantially improved learning. Interestingly, the results indicate that TH feedback, while students complete homework (69% PostTest1), is as effective as receiving no feedback and then having the teacher review of the homework (68% PostTest2). This suggests that to save time, teachers may not even need to review the homework if students have access to web-based homework systems.

3.3 Observational Results

In addition to examining the effects of immediate feedback on learning, this study explored the potential changes to the homework review process the following day in class. In the traditional format of homework review, time must be spent first on checking answers and then the teacher responds to students' questions. However, we hypothesized that when teachers have access to the item report they are able to identify common misconceptions and address those ensuring that the time spent reviewing homework is meaningful.

Remember, that when reviewing the homework, students were separated by condition. The teacher recorded herself as she reviewed the homework with each group. In the following section we attempt to characterize what happened in the video segments.

As usual, the teacher reviewed the item report in the morning to determine which questions needed to be reviewed in class. The item report (see Figure 1) shows individual student performance as well as class performance at the question level. Common wrong answers are also displayed for each question. Using this information, the teacher noted that triplet 1 showed a common misconception when multiplying powers with like bases. While the item report shows that students learned from the feedback, the teacher still felt it was important to highlight and discuss the error in multiplying the bases of the powers together. Therefore the teacher highlighted question 4.


www.assistments.org/teacher/report/item_level/17493?assignment_id=442557
ASSISTments
Build Tutor Assess Admin Notify School
Account
Classes

Assess › Classes › Pre Algebra 2011-2012 (Aug 03, 2011) › Item/Skill Builder Report

Select Report: Item/Skill Builder Report | Assignment Report | Proficiency Report | Essay Grading | ARRS Reports | Unfinished Assignments Report | Assignment Completions by Date

✗ 80	Student's 1st action was to enter the incorrect answer. Student was given the correct answer.
✗ Scaffold	Student's 1st action was to click on "Break This Problem Into Steps." Student answered correctly without being given the answer.
✗ 10	Student's 1st action was to enter the incorrect answer. Student eventually answered correctly without being given the answer.

Class: Pre Algebra 2011-2012 (Aug 03, 2011)

Section: Entire Class

Assignment: 64 - 4.6 (Negative Exponents) (Problem Set 35511)

Student/Problem [Unanonymize]	Average Data driven	#213205 Data driven	#213206 Data driven	#213207 Data driven	#213208 Data driven	#213209 Data driven	#213210 Data driven	#213211 Data driven	#213212 Data driven	#213213 Data driven
Problem Average	60%	90%	100%	100%	27%	61%	84%	71%	39%	68%
Common Wrong Answers					1/9^10, 56%	1/5^13, 58%		121, 50%	1/15t^11, 33% +feedback	
Correct Answer(s)		1/5^3	1/7^8	1/6^12	1/3^10	1/5^3	1/16^2	1	15/t^11	8/y^4
XXXXX	70%	✓ 1/5^3	✓ 1/7^8	✓ 1/6^12	✓ 1/3^10	✓ 1/5^3	✓ 1/16^2	✗ 121	✗ 3/t^11	✓ 8/y^4
XXXXX	65%	✓ 1/5^3	✓ 1/7^8	✓ 1/6^12	✗ 1/9^10	✗ 5^5/5^8	✓ 1/16^2	✓ 1	✓ 15/t^11	✓ 8/y^4
XXXXX	75%	✓ 1/5^3	✓ 1/7^8	✓ 1/6^12	✗ 1/3^6	✓ 1/5^3	✓ 1/16^2	✓ 1	✓ 15/t^11	✓ 8/y^4
XXXXX	50%	✓ 1/5^3	✓ 1/7^8	✓ 1/6^12	✗ 6561	✓ 1/5^3	✓ 1/16^2	✓ 1	✗ 1/5t^11	✓ 8/y^4
XXXXX	60%	✓ 1/5^3	✓ 1/7^8	✓ 1/6^12	✗ 3^-10	✓ 1/5^3	✓ 1/16^2	✓ 1	✗ 12/t^11	✓ 8/y^4
XXXXX	35%	✗ 5^3	✓ 1/7^8	✓ 1/6^12	✗ Hint requested	✗ 1/5^13	✓ 1/16^2	✗ 0	✗ 1/15t^11	✓ 8/y^4
XXXXX	50%	✓	✓	✓	✗	✗	✗ 162	✗ 121	✗ 1/15t^11	✓ 8/y^4

Figure 2: The item report for the WBH condition as viewed by the teacher. Note that class performance for each question and common wrong answers are provided along with individual student performance.

This was especially important because in triplet 2, students incorrectly applied this concept. Specifically, 39% of students initially got this type of question right (multiplying powers with coefficients and variables). However, learning took place as 68% got the next similar question right. It was therefore puzzling to see that on the third question in that triplet (question number 10), only 45% got the question right. Upon investigating the question, the teacher was able to identify the misconception and therefore addressed it with the class. Students learned in the prior triplet not to multiply the bases together. However, in this problem $(5a^3)(5a^{-5})$ students didn't realize that they should multiply the coefficients, 5 and 5 together. You can see in the video that the teacher highlights the difference between these types of problems.

The third and fifth triplet showed adequate learning. Additionally, questions 1, 2, and 3 were introductory questions and performance was above 90% on each question, therefore the teacher did not feel the need to address any of these questions. Similarly, questions 7 and 20 were challenge questions and were therefore not discussed in class.

However, the 4th triplet proved to be the most challenging and showed little learning. Therefore, the teacher chose to review the first question of the triplet (question number 14.) The teacher was able to identify the common mistakes, which were improperly subtracting the negative exponents as well as dividing the base. Because the next question had the poorest performance on the assignment, the teacher also chose to review question number 15 and highlight the importance of subtracting negative exponents carefully. Performance on this triplet suggests that feedback alone wasn't enough to cause learning. Teacher input and clarification was required.

We designed the experiment with ecological validity in mind. That is to say, we wanted the teacher to naturally review the homework, giving students enough time to ask questions. The hope was that approximately the same amount of time would be spent in each class and by each condition. We were disappointed to find that the classes and conditions varied greatly in the amount of time spent going over the homework. Half of the sections took over nine minutes to review the homework while two of the sections in the TH condition and one in the WBH condition spent substantially less time. This is a threat to the validity of drawing statistical inferences, but given the desire to maintain realistic homework review conditions, these inconsistencies highlight important differences in the homework review methods. We describe these differences in the following sections.

An observational analysis of the video recordings of the teacher reviewing the homework revealed that while the time spent in the WBH condition was often longer than the TH, it was also far more focused than in the TH. Specifically, when students were in the TH condition, on average 1 minute passed before any meaningful discussion took place. Whereas, when students were in the WBH condition, homework review began immediately with the teacher reviewing what she perceived to be the most important learning opportunities.

Figure 3: Video of homework review for experimental condition. To watch the full video, go to: http://www.youtube.com/watch?feature=player_embedded&v=Jb6Szy4fZ2w

Other notable differences in the type of review include the number of questions answered. In the TH condition, 2 classes saw 3 questions each and one saw 7. However, in the WBH condition each class saw 4 targeted questions and 2 classes requested 1 additional question. The variation in question types also is important to note. The teacher was able to ensure that a variety of question types and mistakes were addressed whereas in the TH condition students tended to ask the same types of questions or even the same exact question that was already reviewed. Additionally, students in the TH condition also asked more general questions like "I think I may have gotten some of the multiplying ones wrong." In one TH condition only multiplication questions were addressed when clearly division was also a weakness and similarly, another TH condition only asked questions about division. This accounts for much of the variability in overall review time.

Figure 4: Video of homework review for the control condition. To watch the video go to: http://www.youtube.com/watch?feature=player_embedded&v =tBhcuCnKVCY

In listening to the comments made by students it appears that the discussion in the TH condition was not as structured as the WBH condition. Not all students had their work and therefore couldn't participate in the review. One student said, "I forgot to write it down." Another said, "I left my work at home." Because students were asking questions and the teacher was answering them, we suspect that only the student who asked the question was truly engaged. In fact, one student said, "I was still checking and couldn't hear" which led to the teacher reviewing a question twice. In the WBH condition, the teacher used the information in the report, such as percent correct and common wrong answers to engage the entire class in a discussion around misconceptions and the essential concepts from the previous question.

Other notable differences include the completeness of the review. In the TH condition, the review was dominated by student directed questions. This means that each class experienced a different review and the quality of that review was directly dependent on the engagement of the students. Conversely, in the WBH condition, all 3 classes were presented with the same 4 troublesome questions and common mistakes. Additional questions were reviewed when asked (as in two sections) but the essential questions as determined by the data in the item report were covered in all three sections.

3.4 Student Survey Results

Following participation in this study, students were questioned about their opinions. We want to acknowledge that students might have been telling the teacher what she wanted to hear: the whole classroom of students had been using ASSISTments for months and the teacher had told them on multiple occasions why it's good for them to get immediate feedback. So with that caveat, we share the following results. 86% of students answered ASSISTments to the question "Do you prefer to do your homework on ASSISTments or a worksheet?". 66% mistakenly think that it takes longer to complete their homework when using ASSISTments (we showed in this study that that was not the case) and 44% feel that they get frustrated when using ASSISTments to complete their homework. However 73% say that their time is better spent using ASSISTments for their homework than a worksheet. When asked what students like best about ASSISTments, student responses included:

"Being able to try again."

"That if you get stuck on a problem that it will give you the answer."

"You can redo your answer if you get it wrong and learn from your mistakes."

"How it tells you immediately that you are right or wrong."

"I like how I know if I'm right or wrong. This helps because often times when I get things wrong I just go back to my work and I see what I'm doing wrong which helps me when doing other problems."

"I like knowing if your right or wrong. it helps me learn from my mistakes because it makes me go back and keep trying until I get it right. I cant just move on when I feel like it. normally I would just try it a 1st time, and not go back and check, but assistmsnt makes me double Check my work."

"My favorite thing about ASSISTments is that it will tell you if you get the question wrong. PS--it doesn't help when it just says you get it wrong, it's helpful to see the steps so you can compare it to what your answer looked like."

"I like that you can tell what you did wrong and learn from it. That's it though. otherwise I would prefer a wkst [worksheet]."

"I like how it is online and easy to access."

While the learning benefits are profound and students prefer a web-based system, there is a sense of frustration that must still be addressed. Specifically, when asked what should be changed about ASSISTments, student responses included:

"I would make the hint button give a hint and not just the answer."

"I would make it so the hints maybe give you another example or helpful information so instead of just getting the answer and not knowing how you got it you could actually learn from it."

"If you get it wrong more than 4 times you have to move on to the next question."

"I would change how long it takes you to type it in. it would be cool if you could just say the answer and it would enter it in. that probably won't happen, but it would be awesome."

"I would change it to having hints to tell you if you have a little mistake when you hit submit answer so you don't get it wrong because of that little mistake."

This feedback suggests that students appreciate the features of intelligent tutoring systems, including hints, worked examples and scaffolding. Therefore, future studies should explore adding additional feedback to determine if added AIED features improve learning or if maybe learning requires some levels of frustration. All of the survey results are made available without names, including students' comments at http://www.webcitation.org/6DzciCGXm.

4. DISCUSSION

This papers' contribution to the literature is exploring the potential use of ITS for homework support. Used as designed, ITS are somewhat cumbersome for teachers to use for homework as the content is not customizable. However, if ITS are simplified they could be used like web-based homework systems, providing correctness feedback to students and reports to teachers. This begs the question, is correctness only feedback enough to improve the efficacy of homework and what effect does teacher access to reports have on homework review? This randomized controlled study suggests that simple correctness-only feedback for homework substantially improves learning from homework. The benefit of teachers having the data to do a more effective homework review was in the expected direction (but not reliable). But taken together (immediate feedback at night and an arguably smarter homework review driven by the data) the effect size of 0.56 seems much closer to the effect of complex ITS. Of course the large 95% confidence interval of [0.07 to 1.08] tells us we need more studies.

Future studies can explore features of other web-based homework systems like Kahn Academy to determine which aspects of the systems are particularly effective. Incrementally adding tutoring features to determine the effectiveness of each feature would also be valuable. Finally, the role of data in formative assessment should be further explored. In what way can teachers use the data to improve homework and review and instruction?

Caveats: the participants in the current study were all advanced middle school students. Therefore it would be necessary to replicate this study across a broader range of student abilities to determine if these effects are generalizable. Additionally, the correctness feedback is confounded with the unlimited attempts provided on the homework assignment. Therefore, it would be interesting to see if it's simply the correctness feedback that contributes to learning or if the impact stems from the unlimited attempts to self-correct. Finally, to address the secondary research question of the effectiveness of using that data and item report to enhance homework review, a more complicated research design would be required. Specifically, in the present study, the effect of the homework review was confounded with already improved learning that resulted from having correctness feedback. A two-by-two design where both immediate feedback and the factor of going over the homework with the data varies would be necessary.

In this fast-paced educational world, it is important to ensure that time spent in class and on homework is as beneficial as possible. This study provides some strong evidence that web-based homework systems that provides correctness-only feedback are useful tools to improve learning without additional time.

5. ACKNOWLEDGMENTS

The authors would like to acknowledge support from the Bill and Melinda Foundation via EDUCAUSE as well as IES grants R305C100024 and R305A120125.

6. REFERENCES

[1] Black, P., & Wiliam, D. (2006). Inside the black box: Raising standards through classroom assessment. Granada Learning.

[2] CEM (2013). Accessed 1/28/13 at http://www.cemcentre.org/evidence-based-education/effect-size-calculator.

[3] Cooper, H., Robinson, J. C., & Patall, E. A. (2006). Does Homework Improve Academic Achievement? A Synthesis of Research, 1987–2003. Review of Educational Research. Spring 2006, Vol. 76, No. 1, pp. 1–62.

[4] Kelly, K. (2012). Study Materials http://www.webcitation.org/6E03PhjrP. To browse, see http://web.cs.wpi.edu/~nth/PublicScienceArchive/Kelly.htm.

[5] Kulik, C. C., & Kulik, J. A. (1991). Effectiveness of computer-based instruction: An updated analysis. Computers in Human Behavior, 7, 75–94.

[6] Rochelle (2013). The IES Grant. Accessed 1/28/13 at http://ies.ed.gov/funding/grantsearch/details.asp?ID=1273.

[7] Schneider, S (2012). Accessed 1/28/13 at http://www.iesmathcenter.org/home/index.php.

[8] Shute, V. (2008). Focus on Formative Feedback. Review of Educational Research, 78(1), 153 -189. http://www.ets.org/Media/Research/pdf/RR-07-11.pdf

[9] Singh, R., Saleem, M., Pradhan, P., Heffernan, C., Heffernan, N., Razzaq, L. Dailey, M. O'Connor, C. & Mulchay, C. (2011). Feedback during Web-Based Homework: The Role of Hints In Biswas et al (Eds). Proceedings of the Artificial Intelligence in Education Conference 2011. Springer. LNAI 6738, Pages. 328–336.

[10] VanLehn, Kurt (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. Educational Psychologist, 46(4), 197-221.

[11] VanLehn, K., Lynch, C., Schulze, K. Shapiro, J. A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., & Wintersgill, M. (2005). The Andes physics tutoring system: Lessons Learned. In *International Journal of Artificial Intelligence and Education*, 15 (3), 1-47

Target the controls during the problem solving activity, a process to produce adapted epistemic feedbacks in ill-defined domains.

The case of a TEL system for orthopaedic surgery

Vanda Luengo
Université Joseph Fourier
Grenoble
Vanda.Luengo@imag.fr

Dima Mufti-Alchawafa
Université Joseph Fourier
Grenoble
dima.mufti@gmail.com

ABSTRACT

In this paper we study one feedback process which is adapted to ill-defined domains. Indeed, this process use others aspects than expected solutions to propose a feedback. The feedback process is based in a set of didactical aspects. In particular, the feedback targets the control element of knowledge, i.e. the knowledge that allows to validate one step in the problem solving process. The paper describes the feedback process and its implementation in the framework of one TEL system in orthopedic surgery.

Keywords

Control knowledge, feedback process, ill defined domain, and didactical decision.

1. INTRODUCTION

In ill defined domain one of the challenges is to continue to develop new tutoring strategies and seek out ways to combine existing strategies [13]. This challenge still open in particular when the domain has multiple and controversial solutions or ill-defined task structures [4]. In this framework our research question is how to design a tutoring feedback system which is not only based in defined solutions but in the known characteristics of knowledge and learning situations.

We study one kind of feedback which is adapted and epistemic. It is adapted because it takes into account the individual differences in relation to incoming knowledge and skills among students [18]. It is epistemic because it is specific to the piece of knowledge at stake and its learning characteristics. Compute an epistemic feedback involves knowledge from the learner, the learning situation and the learning domain [11].

We design a process to produce adapted epistemic feedbacks which includes one decisional model based in a set of didactical hypothesis. The process was implemented and tested in the case of orthopedic surgery.

The research discussed in this paper is developed in the framework of the TELEOS[1] platform [9] which is a Technological Enhanced Learning environment for orthopaedic surgery. This platform proposes a set of resources for the student (haptic simulator, online course, clinical case database) and a diagnosis system able to analyse the student productions and make a knowledge diagnosis based in identified controls.

Based in the model presented in this paper we add a feedback system in the TELEOS environment. This implementation proposes a formative feedback which is delayed, i.e. at the end of the exercise in the simulator. The model is presented in the section 4 and the TELEOS example is presented in the section 5.

2. RELATED WORKS

In some domains (like percutaneous screw fixations in orthopaedic surgery) the *knowledge* obtained by experience plays an important role for both the expert teacher and the novice learner during a problem-solving process. This kind of knowledge, often tacit, refers to "work-related, practical know-how that typically is acquired informally as a result of on-the-job experience, as opposed to formal instruction." [22]. This kind of knowledge is pragmatic, obtained by experience. Moreover a skillful learner, even a domain expert, often makes several attempts before arriving at an acceptable solution: the person makes an error and then tries to correct the error several times. Also there are multiple solutions and because some parts of the knowledge are tacit the strategic to a good solution are unclear. This kind of problem is ill structured. Indeed, an ill-structured problem as one that is complex, with indefinite starting points, multiple and arguable solutions, or unclear strategies for finding solutions [19].

Several works address the problem to model ill defined knowledge and build feedback from these models ([13] and [20]). Based in this previous works, Fournier-Vigier et al. [5] pointed the design feedback difficulties in ill defined domains, in particular the difficulties to provide domain knowledge in ill structure problems. All studied paradigms (cognitive task analysis, constraint-based modeling, expert system, data mining algorithms) propose to describe task models using different techniques. The task models could be complete or partial. In all cases the model is used to offer assistance to the learner (ibid. 234). Most of the feedback systems in these approaches try to guide the student to the intended solution, even if it is described partially and beside most of the feedback are goal oriented.

We aim to study a model of feedback that is not only based in calculated solutions. We explore another feedback paradigm which is centered in the validation process more than the attended solution. In others words the feedback will be related to the characteristics of the controls brought into play during the problem solving process: it was brought into play in the right moment? It was valid or invalid? What is its nature ?

We would like to investigate how to produce an adapted epistemic feedback that takes into account these knowledge characteristics and is able to handle the uncertainty coming from

[1] http://teleos .imag.fr

the diagnosis results. Indeed, like more and more intelligent tutoring systems, we chose to use Bayesian network for our diagnostic knowledge.

From adaptive point of view, Shute & Zapata-Rivera [18] propose a four-process adaptive cycle connecting the learner to appropriate educational materials and resources. This four process cycle include (ibid. p 9) *capture* of the information about the learner, *analyze* the information in relation to the learner model, *select* the information for a particular learner and *present* specific content to the student.

In relation to the selection step of the feedback, few systems propose a computer model which describes the decision of a pedagogical feedback following an uncertain diagnosis. Mayo and Metrovic [14] introduce the idea of Pedagogical Action Selection (***PAS***) and identified three general approaches to produce them in intelligent tutoring systems that use Bayesian networks: alternative strategies, diagnostic strategies, and decision-theoretic pedagogical strategies (ibid., p 132).

For us a didactical decision is to propose the best feedback and depending on the diagnosis results. This decision means a choice between different possible hypotheses based on didactical analysis. We use a decision-theoretic approach in order to model this process. The decision-theoretic strategy is used in some ITSs to select tutorial actions that maximize the expected utility. The systems CAPIT [14] and DT tutor [16] use this strategy.

CAPIT is a system for learning capitalization and punctuation in English. To decide two kinds of next feedback (next problem selection, error message selection) this system uses the utility function, which is based on the number of errors that the student made [14]. DT tutor also uses a decisional model: "*For each tutorial action alternative, the tutor computes (1) the probability of every possible outcome of that tutorial action, (2) the utility of each possible outcome in relation to the tutor's objectives, and then (3) the alternative's expected utility by weighing the utility of each possible outcome by the probability that it will occur. The tutor then selects the action with maximum expected utility with utility function*" [16]. In DT tutor, many factors related to the student (their morale, behaviour, etc) have an influence on expected utility. To propose the next feedback, DT tutor chooses first the theme where the feedback is focused and second the type of feedback (help, hint, positive or negative feedback). DT tutor is implemented in two learning systems, calculus and elementary reading.

A further difference between these previous works and our approach is that the decision feedback models proposed previously are not based on the nature of the control knowledge; in our case we would like to center the feedback on the knowledge control dimension (knowledge that allows the users to validate their actions during the process) and to take into account the knowledge control specificities (pragmatic, declarative and perceptive-gestural). Another difference is that, in our learning environment, there are no well defined solutions and thus it is not possible to define a priori, a list of actions as expected feedback. Because we have some characterised resources in our environment, the feedback is built in several steps; it has a target, an objective, a form and content. It is created with a decision-making process based on several PAS (Pedagogical Action Selection). In each step of the process, the chosen strategy corresponds to the degree of dependency of the step in relation to the domain knowledge.

Finally the factors considered in our system must be the parameters that can be established by researcher. Indeed, this is multidisciplinary research that evolves and the system must adapt to the evolution of didactic and medical analysis. Different feedback hypotheses must be able to be tested.

3. THEORETICAL FRAMEWORK AND DIDACTICAL HYPOTHESIS

According to research in cognitive psychology and didactics, the learner/situation interaction can be modelled as a problem-solving process that engages itself different processes, tightly linked and recurrent: identification of the situation, planning, action, control of actions' effects, regulation. The crucial role of control elements in this process has been pointed ([1],[17]), allowing the subject to decide whether an action is relevant or not, or to decide that a problem or sub-problem is solved.

This framework has some important consequences on our work for our objectives related to the design of a feedback system:

- It is necessary to *choose characteristics of problems* that will conduct to the right processes of learning according to professional objectives and to learner's state of knowledge. This, in turn, leads to the necessity to diagnose learner's knowledge, and interpret this diagnosis to be able to provoke targeted learning through learners' actions and controls on problems. Thus, one objective of *the feedback system is to consider* is not only the actions but also *the controls brought into play by the learner during the problem solving activity*.
- It is necessary to *distinguish* and consider both, *the result* (a punctual state of the problem, intermediate or final) *and the problem solving process*. We thus adopt a continuous approach of diagnosis and learning process.

Besides, we argue that is necessary to distinguish the controls characteristics. These categories are related to the way that knowledge can be validated, and therefore, built. In the case of orthopedic surgery we identify three categories: declarative, pragmatic and perceptive/gestural. The first category, declarative knowledge, corresponds to shared knowledge, constituting a common reference for professionals. It can be expressed, formally, and serves communication, discussion, exchanges. The second, pragmatic, is partly expressible, and is linked to individual experiences and situations. The third concern the perceptive and gestural (technical gesture like surgical gesture) part, hardly expressible and embedded in particular situations.

These are not the same that the classical division of knowledge between declarative and procedural. For example, part of procedural knowledge is validated in a declarative manner (is a reference for professionals and transmitted in a declarative way), part is validated in a pragmatic manner (by experience) and can also be validated in a perceptive-gestural manner (what is seen, felt). This second kind of activity is ill defined task, i.e. there are not clear strategies for finding solutions at each step of the problem solving process.

3.1 Characterization of didactical hypothesis' factors

Based in previous framework our objective is to propose a feedback system able to take into account the didactical hypothesis.

First of all and as explained above, each control element of knowledge is labelled according to its nature: declarative, pragmatic, or perceptual/gestural. Then, concerning knowledge related to the user's action, it is labelled according to the moment it appears in the resolution process and according to its possible validity.

This last element necessitates some clarification: knowledge elements are diagnosed by the environment, according to user's set of actions and knowledge domain of validity, as being mobilized (brought into play) in a valid situation state (inside its validity domain), not mobilized or mobilized in an invalid situation state (outside its validity domain).

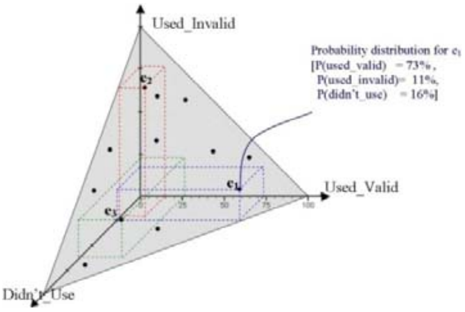


Figure 1. Result of knowledge elements diagnosed

The output can be considered like a tri-dimensional space (shown in Figure 1), where each knowledge element (e_i), in our case controls, has a probability distribution according to their state (invalid, valid, or not-used). In the given example, the knowledge element e_1 is brought into play in a valid manner with a probability of 73%.

Based in this result we made choices concerning the best relevant type of feedback to be provided to the user, according to previous diagnosed elements.

Thus, to produce epistemic feedback, the didactical analysis is based on the characteristics (state, order, type, etc.) of the control knowledge element and the classes of situations available. Also, to integrate the adapted dimension the feedback process has to take into account the student knowledge (the diagnosis result) and the characteristics of the learning environment (resources manipulated by the student and the characteristics of the problem).

4. THE PROCESS TO PRODUCE AN ADAPTIVE EPISTEMIC FEEDBACK

This process has four related steps. First, our decision model chooses the knowledge element(s), proposed by the diagnosis system, which will target the feedback. Second, it determines the feedback's apprenticeship objective for the chosen target. Third, according to the target and the objective, it determines the relevant form of feedback from the existing forms in the learning environment. Finally, according to the form, the decision model formulates the feedback by defining its content. The process is conceived from objectives and didactical hypothesis, summed up in §3, which are represented like parameters in the system.

In the next paragraphs, we describe each step in detail.

4.1 Chose the target of the feedback

This step can be shown as the selection of knowledge elements which are target by the feedback. The selection is influenced essentially by the knowledge diagnosis results and the controls' characteristics. In our case the knowledge elements are the controls which are brought into play during the problem solving activity. At each student action a list of controls were diagnosed. The results of one step can be seen like in the Figure 1. This diagnosis system is described in Chieu et al. [4].

We use influence diagrams to represent this step of decision. It is used to represent and to calculate the decision-making in several applications [6], [7]. In the influence diagrams there are decision nodes and utility nodes as well as chance nodes.

We have chosen this approach because it allows making decisions under uncertainty. Indeed, the learner's state of knowledge, produced by the diagnosis, will be deduced from learner actions with a degree of uncertainty, so our model has to generate the best feedback according to this input.

In our model (Figure 2) there are knowledge nodes (the oval nodes that represent the result of the diagnosis), an apprenticeship utility node (hexagonal node) and target decision node (rectangular node with the list of candidate elements or knowledge to be targeted). The inference in this diagram allows selecting a knowledge element as target. Indeed, the result of the inference gives the values of the utilities for each knowledge element, the highest one will be the targeted element for the feeback.

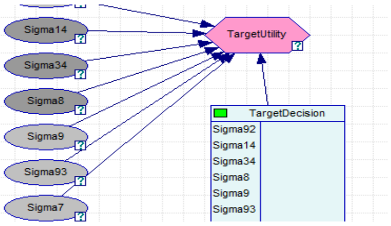


Figure 2. The influence diagram for target decision

To apply the inference in the diagram, we defined a function that models the preferences from an apprenticeship point of view, which is the utility function. The preferences will be described numerically under the notion of utility U, where U(a1)> U(a2) means the decision-maker prefer action a2 compared to the action a1.

In our case the apprenticeship utility function, Uapp(c, E), allows us to calculate the a priori utility to focus feedback on an element knowledge of a candidate (c) by taking into account the set of knowledge elements (E). Then, the inference in the influence diagram calculates the estimated utility for each candidate according to the diagnosis results. In other words, the utility function initializes the calculation in the influence diagram and then the inference algorithm deduces the decisions.

As we can see, in the previous figure the diagram is very simple; our contribution is basically in the definition of the apprenticeship utility function that takes into account the didactical hypothesis, which we explain in the next paragraphs

4.1.1 Apprenticeship utility function

This utility function allows initializing a priori utilities according to the factors that influence the target decision. We identified some factors as the element state and the element characteristic:

1. Element State is the diagnosis result. It represents the manner of using the knowledge element in the problem-solving process: Used-valid, Used-invalid, not-used.
2. Element Type, it is linked to the validation criteria for each identified knowledge, like explained after, in our

current Teleos example it can be “declarative”, “pragmatic” or “perceptive-gestural”;

3. Element Order represents the step of problem-solving in which this element intervenes. An element can intervene in several problem-solving steps, for example the control knowledge related to the profile X-ray can intervene in several steps of the activity;
4. Element Context indicates the context of problem-solving in which this element intervenes. It can be ‘general’ or ‘particular’. For example, in the case of surgical domain, some steps and knowledge control elements could be especially for the scoliosis intervention.

From all of these factors we define in the equation (1) $U_{app}(c, E)$ the utility to choose a candidate element, ***c,*** as feedback target in taking into account the set of knowledge elements, ***E***, as the sum of all the utilities related to each factor.

$$U_{app}(c,E)=\alpha\cdot U_{state}(c,E)+\beta\cdot U_{Type}(c)+\gamma\cdot U_{order}(c)+\delta\cdot U_{context}(c) \quad (1)$$

In our didactical hypotheses, these factors do not have the same weight in influencing the choice of the target. Thus, we attribute to each factor a priority variable (α, β, γ, and δ), which represents its weight in the utility calculation.

We define in the equation (2) the utility of choosing a candidate ***c*** as a target according to its state $U_{state}(c, E)$, as the sum of utilities for each pair of candidates ***c*** and element $\boldsymbol{e_j}$ in ***E***; n is the number of knowledge elements of the set E.

$$U_{state}(c,e_1,e_2,...,e_n)=\sum\nolimits_{j=1}^{n}U_{State}(c,e_j) \qquad (2)$$

In addition, we define the state utility in the table for each pair $U_{state}(c, e_j)$. The values are defined according to didactical hypotheses and the domain of knowledge.

For example, the didactical hypothesis “*it is more important to focus the feedback on an element that is used in an invalid way than to focus it on an element that it didn’t use”* is represented by a value where Ustate(c = “used_valid”, e) ≥ Ustate(c = “not-used”, e). In other words, we propose one utility state table that allows selecting between two elements situated in the diagnosis results space (shown in Figure 1) according to the chosen didactical hypothesis.

The definition of the type utility $U_{type}(c)$ from didactical hypothesis can be “*it is more important to focus the feedback on a declarative element than to focus it on a pragmatic one*”. We express this by giving to declarative elements the higher value of utility. In this example, the $U_{type}(c) = 3$ if c is declarative and 2 if it is pragmatic. In the present implemented version, the system doesn’t take into account the perceptive-gestural knowledge because the didactical analysis is ongoing, but it is modelled to integrate it in an easy and modular manner.

We define the utility order: Uorder(c), from the didactical hypothesis “it is more important to focus the feedback on an element appearing in a primary stage of the solving than to focus it on an element appearing in later stages”. Thus, it is possible that an element appears in several stages. We define the utility order in equation (3); m is the number of steps where this element appears and O(c) is its order. The first time of the control *i* is identified Oi(c) = 1.

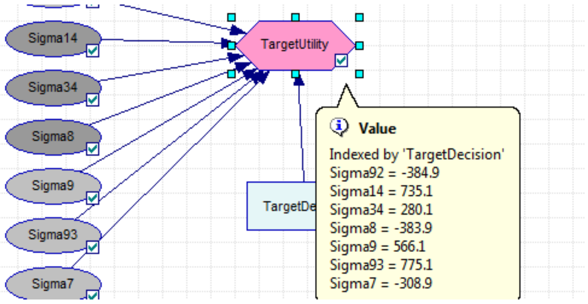
$$U_{order}(c)=\sum\nolimits_{j=1}^{m}\frac{1}{O_j(c)} \qquad (3)$$

We define the nature utility $U_{nature}(c)$ from the didactical hypothesis as follows: “*it is more important to focus the feedback on an element appearing in the solving of a general problem than to focus it on an element appearing in a particular context*”. Like the Utility type function case, we express this by giving a higher value of utility to the nature target chosen (in this case if c is general $U_{context}(c)= 2$).

According to these considerations, we have defined an algorithm that calculates the apprenticeship utility function and initializes the utility table from a set of knowledge elements with their characteristics. In this algorithm we create, first of all, the coefficients’ matrix «*coeff*» in relation to the number of knowledge elements (k), and then we calculate the state utility table for each candidate. It is calculated based in formula 4, where *k* is the number of the column, j is the possible state of the knowledge element (used-valid, invalid or not-used) and *Hypo* is one of the didactic vectors A,B or C related to the state of the targeted candidate in column *k*

$$ValeurUtilitéEtat[k]=\sum_{j=1}^{3}Coeff[j,k]*Hypo[j] \qquad (4)$$

This algorithm needs to be to running only once, after settle the didactical hypothesis. The inference in the influence diagram then uses probabilities resulting from the diagnosis and then calculates utility values to infer the estimated utility for each element. Finally, the target for the feedback is the element that has the maximal estimated utility value (Figure 3) calculated. It is possible to have some elements with the same maximal utility.



Value
Indexed by 'TargetDecision'
Sigma92 = -384.9
Sigma14 = 735.1
Sigma34 = 280.1
Sigma8 = -383.9
Sigma9 = 566.1
Sigma93 = 775.1
Sigma7 = -308.9

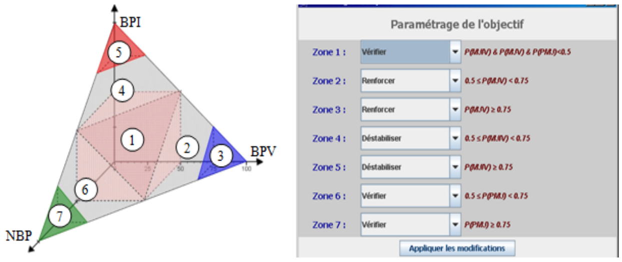
Figure 3. Inference Diagram decision result

As we presented before, we have chosen to represent all didactical hypotheses as parameters in the utility function. This choice makes our model flexible to add or modify didactical hypotheses. For example, for the factor “Type of Knowledge” if the didactical hypothesis is “*it is more important to focus the feedback on an pragmatic element than to focus it on a declarative one*”, then it is sufficient to give the parameter that represents the utility for an pragmatic element a value higher than the utility for a declarative element $U_{type}(c=pragmatic)> U_{type}(c=declarative)$.

4.2 Choose the objective of the feedback

After choosing the target, the decision model determines its feedback objective in order to give, from the learning point of view, a semantic to the feedback intention. In our model we distinguish several feedbacks. Indeed, if the target knowledge is diagnosed (with a higher probability) as ‘brought into play in an invalid manner' (BPI) the feedback is not the same than if this target knowledge is diagnosed as 'not brought into play' (NBP).

We have defined a procedure that determines the feedback objective by applying an analysis on the target element state. The principle of this procedure is that it segments the diagnosis space into several zones, and it attributes an objective to each zone. Then, the feedback objective corresponds to the zone in which the target element is situated. This step permits to pass from an uncertain state of knowledge to fixed objectives of learning. The number of segmented zones and the objective for each zone is customizable in our model.



Paramétrage de l'objectif

Zone 1 : Vérifier P(M,IV) & P(M,V) & P(PM,I)<0.5
Zone 2 : Renforcer 0.5 ≤ P(M,V) < 0.75
Zone 3 : Renforcer P(M,V) ≥ 0.75
Zone 4 : Déstabiliser 0.5 ≤ P(M,IV) < 0.75
Zone 5 : Déstabiliser P(M,IV) ≥ 0.75
Zone 6 : Vérifier 0.5 ≤ P(PM,I) < 0.75
Zone 7 : Vérifier P(PM,I) ≥ 0.75

Figure 4. Example of segmentation of the knowledge elements space to determine the feedback objective

In this example, if the knowledge element is in zone 1 ("*if P(NBP) – P(BPI) > 0.25 and P(NBP) – P(BPV) > 0.25*") then the feedback objective is to "verify" if the targeted knowledge is understood by the learner. The possibilities proposed for the feedback objective are: verify, reinforce and destabilize. The meaning of *verify* feedback is to propose a type of feedback to improve the diagnosis related to a set of knowledge targeted elements (for example, proposing another problem where specific targeted knowledge has to be mobilized). The idea of the *reinforce* feedback is to support the user in relation to the targeted knowledge elements (for example a positive feedback, a closer clinical case that was studied or solve another problem where the targeted knowledge could be used). Finally, *destabilize* feedback has the objective to show that the targeted knowledge is used in an invalid manner in these kinds of problems (by explaining the right way in the course, proposing a counter example from the clinical case database or proposing another problem where if the knowledge is used, the result could be wrong)

4.3 Determine the feedback form

In this step, the decision model chooses the most relevant form of feedback linked to the type of the target knowledge element and the feedback objective (*reinforce, verify, destabilize*).

Here the idea is to associate one kind of feedback form to the feedback objective and the type of the targeted knowledge element. In this step we need to consider the resources proposed to the student. Indeed more and more TEL system proposes several resources to the student. For example if the environment has a wiki with concepts we can associate it to a form of feedback when the targeted knowledge element is declarative and the feedback objective is to reinforce.

This association is a simple table where we can match the resources with a pair <type of knowledge, feedback objective>.

4.4 Determine the feedback content

The content is essentially related to the form of feedback. Here the objective is to determine the content of the feedback in relation to the feedback form. For example if the feedback form is a wiki with concepts the content has to be related with the targeted knowledge element.

This step is not generic, it depends on the kind of feedback forms that the TEL system has. For this reason this step will be more detailed in the next section where we explain the case study where we implemented the feedback process.

5. THE TELEOS SYSTEM EXAMPLE

The analyzed procedure is about surgical orthopaedic percutaneous (without incision) operation. It is developed in [21]. It could be summarized as follows: The surgeon first inserts a pin in the bone through the skin. S/he makes the pin progress in the bone, taking several X-rays to validate the pin's course at different steps of its progression. The X-rays allow him or her to "reconstruct" a complete vision of the position of the pin, in relation to the bone. If s/he recognizes any problems in those views, s/he restarts the operation process, taking another pin and correcting its entry point and/or direction. Until now we have analyzed the sacroiliac screw operation and the vertebroplasty. The description procedure does not have to be complete and well-defined but the goal is to extract from the diversity of each particular situation, the significant controls elements, from a learning point of view, of the surgical procedure.

The analysis, made in [21], allows us to identify crucial aspects of the surgical procedure. We identified primarily that the pin's positioning is the most important part of the procedure, the definitive screw being placed along this pin. Secondly, we notice the crucial role of X-ray controls. As the surgeon cannot directly visualize the operating area, he has to interpret his gesture through these controls. This necessitates two levels of interpretation. On the first level, the surgeon has to ensure that the X-ray is valid (i.e. being oriented in order to represent what it is intended to represent); on the second level, the surgeon can look at the validity of the pin's position according to anatomical criteria on the X-ray.

Table 1. Examples of knowledge controls for sacroiliac screw

Control Type	Control elements of knowledge	Domains of validity
declarative	The pin's trajectory must be completely intra-osseous	all
declarative	If the pin is well positioned then the pin appears as a point on the profile X-ray	PB, PC, PE
Pragmatic	If the pin would become extra osseous by being pushed in S1, 1cm after the median line, then it can be stopped at the median line	PC, PD
Pragmatic	If the pin would become extra osseous, then it can be stopped just 1cm after having reached S1	PA,PD,PF
Perceptive-gestural	If the pin was in the sacroiliac and the resistance force decrease then the pin would become extra osseous	All

Thus, we identified the control knowledge elements, which are related to surgeons' actions during the intervention, they allow surgeons to validate their actions; some examples are shown in Table 1. The controls have a domain if validity, i.e. they are valid for a set of problems. The control type is also identified: it could be declarative, pragmatic, or perceptive gestural.

5.1 TELEOS SYSTEM

We have developed a modular architecture. Each module is built in relation to the knowledge learning constraints [10]. The learner interacts with the following modules: Semantic Web Courses, Simulator, and Clinical Cases. We introduce briefly these three modules in the next section. The decision-making model uses these modules and the result of the diagnosis to build the feedback. The diagnosis model will not be described in this paper. The result will show in the Figure 1.

5.1.1 *Simulator for orthopaedic surgery*

The last implementation version is explained in a previous paper [12]. Two surgeries were implemented in this last version: the vertebroplasty and the sacroiliac screw.

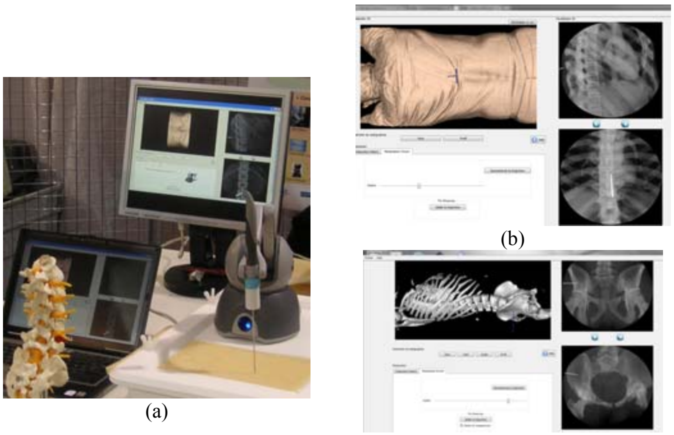


Figure 5. Haptic interface (a). Graphical interface during the pin trajectory (b). Graphical interface when the trajectory is validated by the user (c).

Regardless of the simulated operation, the TEL system gives to the learner the opportunity to train himself and practise a surgical operation thanks to several functionalities: Choosing the type of patient and the type of operation; visualizing in 3D the tool and the patient model; Adjusting the position and the incidence of the fluoroscopic image intensifier; Drawing the cutaneous marking on the body of the patient model; Producing and visualizing radiographies; Manipulating the surgical tool through a mouse or through haptic interface; Verifying the trajectory in a 3D bone model when it has been validated (Figure 5). In this paper we are focused on the pelvis operation.

In the previous figure we can see on the right of the graphical interfaces (b and c), two 2D images representing the last two radiographies produced by the user. In the top left hand corner, there is the 3D model of the patient, and the surgical tool, the user is able to see the 3D bone model only when the trajectory is validated.

5.1.2 *Clinical cases database*

The role of the Clinical Case agent is to illustrate the consequences of a proposed trajectory. It is a database where we can find pertinent information related to different phases (before, during and after the operation).

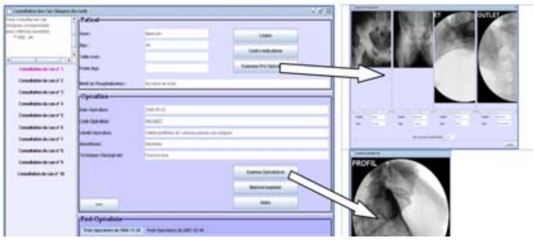


Figure 6. Clinical Case with data from one operation

For example, one clinical case may have some x-rays before the operation (**Figure 6**, right side), some films of the gesture during the operation and some x-rays and data describing the post-operatory information (the position of the bone, the state of the bone, etc… left side **Figure 6**). This Clinical Case Database could be useful to show, for example, trajectories that have consequences in the post-operatory period (there may be a problem with the fracture reduction because the trajectory with the pin is too short, for Instance).

5.1.3 *Online Courses*

We have an online course (at http://www-sante.ujf-grenoble.fr) that explains the declarative knowledge (anatomy, surgical procedure, tools, etc.) about sacroiliac percutaneous screw placement. It is based on online courses and academic documentation, and is improved by interaction between the didactical expert and the surgeons.

For this part we use ontology with a set of rules based in OWL language. We have developed a semantic web module, with more than eighteen web pages, which have metadata based on ontology. This module proposes not only syntactic links, but also semantic ones; it allows the redirection to precise and relevant chapters of the online course. The implementation of this module is explained in previous work [8].

5.2 ADAPTIVE AND EPISTEMIC FEEDBACK PROCESS

Like introduce in the paper the implemented feedback process is a delayed feedback, i.e. the TELEOS system propose a feedback at the end of the activity. The result of the process can be to solve another problem on the simulator, to consult a particular webpage on the online course or to consult one specific clinical case in the database.

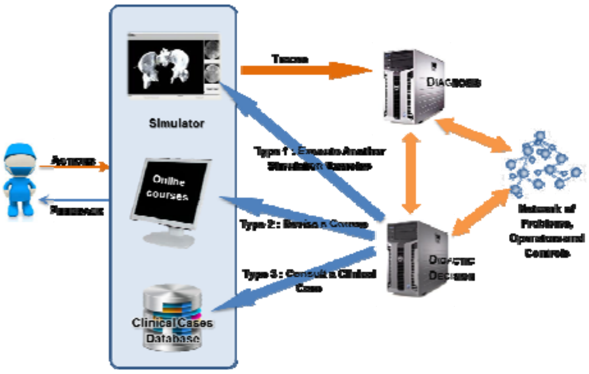

Traces
Diagnosis
Simulator
Actions
Feedback
Type 1 : Execute Another Simulation Exercise
Online courses
Type 2 : Revise a Course
Type 3 : Consult a Clinical Case
Didactic Decision
Network of Problems, Operators and Controls
Clinical Cases Database

Figure 7. Kinds of feedback in TELEOS system.

Because the two first steps described previously are generic, we don't explain them in detail here. In the step three we propose a simple table interface where the didactical or pedagogical user can propose the match between the resource (simulator, clinical case and web course) and the pair type of knowledge (declarative, pragmatic, perceptivo-gestural) and feedback objective ((*reinforce, verify, destabilize*). We can choose one or several forms for the same pair <type, objective>. For example, the pedagogical user was proposed the clinical case and the simulator to destabilize the pragmatic knowledge.

For the step 4 we need to consider the specific form of the feedbacks. In our case we have three forms of feedback (online web course, simulator, and clinical database) and to find the inquired form we do not apply the same process. One example of possible feedback is shown in the next figure:

In the case of the form 'consult part of web courses', the content represents the links to the appropriate pages. It is made by sending keywords related to the target element to a semantic web model [8]. This feedback receives the knowledge elements to be considered, which are analyzed by the java program, using the ontology, and finally it produces a web page with a set of links to the online course, which are related to the targeted knowledge. The Java Engine code uses the open source tool Jena which offers libraries to work with OWL files. In the case of the sacro-illiac surgical operation, our system is based on two ontologies, one related to the pelvis anatomy, which is built on Standford university anatomy ontology [2], and the other one is related to the screw placement procedures, which we built and validated with our experts.

For example if we give the knowledge element 'Outlet radio control', which is in relation with anatomical ontology, the java engine finds the classes related to these knowledge elements and produces a set of links which come from the online course.

The calculation of the content for the forms 'clinical cases' and 'simulator' is made according to the target and to the feedback objective. For the form 'consult a clinical case', it represents the relevant case like a query in a database.

Finally, for the form 'solve another problem with the simulator', it represents the relevant problem to solve. The design can be made by applying inference algorithms in the Bayesian Network (that represents the knowledge domain) or by a decisional theoretical approach to select a closer problem [15]. In the present version we find the problem that has the most common didactical variables (kind of fracture, the hardness of the spongy bone) with the solved problem.

5.3 Evaluation and discussion

The evaluation of the didactical decision process was achieved in several steps. Because the utility function is additive, we evaluated first the dominance between different modelled factors and second we made a sensitive analysis to study the adaptability of the model. Moreover, we made an evaluation to study the behaviour of the system in relation to the expert's propositions. Here we present this last evaluation. The others evaluations show that, firstly, small changes in the assigned probabilities lead to different decisions of feedback target. It means that if there is one small change then the result of the calculus of the target feedback could be radically different. Secondly, the sensitivity level can be adjusted according to the weight given to the element state factor.

The aim of the comparison with expert proposition is to verify and refine the model in relation to the human didactical feedbacks. Here the input is the simulated diagnosis of learner's state of knowledge (e1 [BPI 0.7, BPV 0.17, NBP 0.13], e 92 [BPI 0.65, BPV 0.23, NBP 0.12], etc) and the output is the feedback proposed (Consult the parts of the course 'entry point related to skin marks', propose a problem, with a disjunction, to solve in the simulator, etc.). These scenarios are run by an expert in didactics and by our didactical decision system, afterwards they are compared.

Because in our model the didactical hypotheses are customizable, the parameters have to be calibrated by an expert (in didactics for example) before using it. To make the adjustment of these parameters easier, we developed some interfaces and we also proposed a questionnaire that contains multiple-choice questions, (associate to didactical hypothesis) and we associate with each choice a possible value of the parameter. Therefore, the answers to this questionnaire allow initializing the calculation in the model.

One example of scenario given to the experts is "after radio outlet, a student does not takes Inlet radio and modifies its trajectory in the wrong direction (the pin was placed a little low on the outlet, it starts and moves the point of entry down). The declarative control e93 (coupling outlet / inlet) comes NBP 30%, the declarative control e19 (risk of passing through the hole of the sacrum because too low on outlet) is BPV 50% and the pragmatic control e18 (link outlet position / position of patient) is 75% BPI". One expert proposition was: "propose the web page linked to the inlet/outlet coupling and propose an exercise related to the 2D and 3D association".

In relation to the configuration of the system, the answer of the questionnaire shows us a dependent relationship between the state of the knowledge elements and its characteristics while in our model these factors are independent (it is an additive function). For example, the question about what is more important to target a "not-valid knowledge" or a "not used knowledge", the expert answer depends on the type of the knowledge (declarative, pragmatic, etc.).

In addition, regarding the output proposed by the expert, the results show that the system is able to produce relevant feedbacks for each scenario. Furthermore, some feedbacks are not exactly the same as the expert feedbacks. We identify two reasons for these differences. Firstly, the present model selects as target one (or some) element(s) that has(have) the maximal value of estimated utilities but in the expert propositions, the feedbacks can be related to some elements with positive values of estimated utilities and related as well to the elements with the maximal value. Secondly, the present model is not able to propose a sequential set of feedbacks (for instance, the expert proposes that feedback 1 follows feedback 2). In fact, the present model is able to take the historical dimension with the evolution of the probabilities, but it does not yet treat the historical dimension related to the previous feedback

6. DISCUSSION

This system had to support an explicit representation of pedagogical and didactical hypotheses and, from a computer architecture point of view; the system had to be separated from the other modules. These choices are related to the idea of proposing a normative system, able to evaluate separately and also to allow the investigation of some didactical hypothesis to generate the feedback.

The decision model thus integrates didactical hypotheses in order to represent the decision-maker's preferences. These didactical hypotheses are customizable; this choice makes our model dynamic and partially generic. Also, this kind of model intends to allow multidisciplinary work in order to investigate pedagogical feedback.

From the epistemic dimension of the feedback point of view, the system cannot be completely generic but the design allows identifying the generic steps from the knowledge analysis dependant steps.

In relation to the adaptive dimension of the feedback, the system is able to adapt the feedback to some epistemic considerations about the user and the available resources. Indeed, this adaptive dimension takes into account only the knowledge factors. It doesn't take into account other factors like the morale or attention. Also, as pointed out by Woolf ([23] p. 133), it is necessary to integrate different teachers' strategies: *A single teaching strategy was implemented within each tutor with the thought that this strategy was effective for all students. However, students learn at different rates and in different ways, and knowing which teaching strategy (...) is useful for which student would be helpful. This section suggests the need for multiple teaching strategies within a single tutor so that an appropriate strategy might be selected for a given student".*

The reliability of our model depends on the accuracy of diagnosis results and the best set of parameters. Here it is also necessary to refine the model using real data in order to improve its structure, the conditional probability and the decision factors by using a method of automatic learning from data.

Moreover, the evaluation indicates that it seems necessary to consider not only the history of the student activity but also the dynamic aspect linked to the decisions. Indeed, in the classical approach the decision is in relation to the predictive aspect of the student model ([16], [2]) i.e. it calculates the consequences of the feedback on the predictive student model. However, it appears that the dynamic aspects concern not only the student factors but also the resources or the decision itself.

The data collection seems to be the perspective's keystone in order to improve the present model but also to go forward in this kind of research. However, the data to be collected it is not only the classical data in the domain of learning systems, i.e. the data from the student, but also the data linked to the feedback decision. This kind of collection will be more centred on the analysis of the decision process for the feedback production.

7. REFERENCES

[1] Balacheff N., Gaudin N. (2010) Modeling students' conceptions: The case of function. *Research in Collegiate Mathematics Education*, Volume 16, 183-211.

[2] Chieu, V., Luengo, V., Vadcard, L., & Tonetti, J. (2010). Student modeling in complex domains:Exploiting symbiosis between temporal Bayesian networks and fine-grained didactical analysis. *Journal of Artificial Intelligence in Education..*

[3] Foundational Model of Anatomy, S. (2006). *Research Projects, Foundational Model of Anatomy*. Retrieved 2008, from http://www.smi.stanford.edu/ (Research Projects, Foundational Model of Anatomy).

[4] Fournier-Viger, P., Nkambou, R. & Mephu Nguifo, E. (2010). Building Intelligent Tutoring Systems for Ill-Defined Domains. In Nkambou, R., Mizoguchi, R. & Bourdeau, J. (Eds.). *Advances in Intelligent Tutoring Systems*, Springer, p.81-101.

[5] Fournier-Viger, P., Nkambou, R., Mayers, A., Mephu Nguifo, E & Faghihi, U. (2012). Multi-Paradigm Generation of Tutoring Feedback in Robotic Arm Manipulation Training. *Proceedings of the 11th Intern. Conf. on Intelligent Tutoring Systems* Springer, pp.233-242.

[6] Horvitz, E., Kadie, C., Paek, T., & Hovel, D. (2003). Models of attention in computing and communication: from principles to applications. *Commun. ACM* , 52-59.

[7] Kabanza, F., Bisson, G., Charneau, A., & Jang, T. (2006). Implementing tutoring strategies into a patient simulator for clinical reasoning learning. *Artificial Intelligence in Medicine* , 79-96.

[8] Luengo, V., & Vadcard, L. (2005). Design of adaptive feedback in a web educational system. *Workshop Adaptive Systems for Web-Based Education: Tools and Reusability*. In International Conference on Artificial Intelligence in Education. Springer-Verlag.

[9] Luengo, V., Vadcard, L., Dubois, M., & Mufti-Alchawafa, D. (2006). TELEOS : de l'analyse de l'activité professionnelle à la formalisation des connaissances pour un environnement d'apprentissage. *IC 2006.*

[10] Luengo V. (2008) Take into account knowledge constraints for TEL environments design in medical education. *ICALT* 2008.

[11] Luengo V. (2009). *Les rétroactions épistémiques dans les Environnements Informatiques pour l'Apprentissage Humain. Habilitation à diriger des recherches.* Habilitation à diriger des recherches. Université Joseph Fourier., 2009.

[12] Luengo V., Larcher A., Tonetti J.. (2011) Design and Implementation of a Visual and Haptic Simulator in a Platform for a TEL System in Percutaneuos Orthopedic Surgery. *Medicine Meets Virtual Reality* MMVR, 2011.

[13] Lynch, C., Ashley, K., Aleven, V., & Pinkwart, N. (2006). "Defining Ill-Defined Domains; A literature survey." *Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains at the 8th International Conference on Intelligent Tutoring Systems* (p. 1-10).

[14] Mayo, M., & Mitrovic, A. (2000). Using a Probabilistic Student Model to Control Problem Difficulty. *Intelligent Tutoring Systems* (pp. 524-533). Biarritz: Springer-Verlag.

[15] Muldner, K., & Conati, C. (2007). Evaluating a Decision-Theoretic Approach to Tailored Example Selection. *IJCAI*, (pp. 483-488).

[16] Murray, R., VanLehn, K., & Mostow, K. (2004). Looking ahead to select tutorial actions: A decision-theoretic approach". *International Journal of Artificial Intelligence and Education* , 235-278.

[17] Schoenfeld, A. (1985). *Mathematical Problem Solving*. New York, NY: Academic Press.

[18] Shute, V. J., & Zapata-Rivera, D. (2012). Adaptive educational systems. In P. Durlach (Ed.), *Adaptive technologies for training and education* (pp. 7-27). New York, NY:Cambridge University Press.

[19] Simon, H. A., "The Structure of Ill Structured Problems," *Artificial Intelligence* **4**, (1973), 181-201.

[20] Stamper, J., Barnes, T., and Croy, M. (2010) Enhancing the Automatic Generation of Hints with Expert Seeding. Proceeding of ITS2010. vol. II, pp. 31-40. Berlin, Germany: Springer Verlag.

[21] Tonetti J., Vadcard L., Girard P., Dubois M., Merloz P., Troccaz J. (2009). Assessment of a percutaneous iliosacral screw insertion simulator. *Orthopaedics & Traumatology: Surgery & Research, Vol 95* n°7, 471-477.

[22] Wagner, R., Sujan, H., Sujan, M., Rashotte, C., & Sternberg, R. (1999). Tacit Knowledge in Sales. In S. R. J., & H. J. A., *Tacit Knowledge in Professional Practice: Researcher and Practitioner Perspectives* (pp. 155–182). Mahwah, NJ: Lawre.

[23] Woolf, B. P. (2009). Building Intelligent Interactive tutors : student-centered strategies for revolutionizing e-learning. Burlington, MA, USA: Elsevier

Eliciting student explanations during tutorial dialogue for the purpose of providing formative feedback

Pamela Jordan
Learning Research and Development Center
University of Pittsburgh
Pittsburgh PA, USA, 15260
pjordan@pitt.edu

Patricia Albacete
Learning Research and Development Center
University of Pittsburgh
Pittsburgh PA, USA, 15260
palbacet@pitt.edu

Michael J. Ford
School of Education
University of Pittsburgh
Pittsburgh PA, USA, 15260
mjford@pitt.edu

Sandra Katz
Learning Research and Development Center
University of Pittsburgh
Pittsburgh PA, USA, 15260
katz@pitt.edu

Michael Lipschultz
Department of Computer Science
University of Pittsburgh
Pittsburgh PA, USA, 15260
mil28@pitt.edu

ABSTRACT

In this paper we explore the question of whether additional benefits can be derived from providing formative feedback on students' explanations given the difficulties of accurately assessing them automatically. We provide a preliminary evaluation of an approach in which students assist in interpreting their own explanations and we lay out our plans for evaluating the effectiveness of a natural-language intelligent tutoring system's feedback to that interpretation effort. The preliminary evaluation suggests that students respond well to the approach. While their interpretation assistance may be similar to an automated explanation matcher, they continue to provide explanations throughout their interactions.

Keywords

student explanations, tutorial dialogue, formative feedback

1. INTRODUCTION

Numerous studies suggest that self-explanations can be more beneficial to students than explanations from others (e.g. [3]). In the context of an automated learning environment this raises the question of whether additional benefit can be derived from providing formative feedback on any explanations the student enters when the automated understanding of those explanations remains a major obstacle. Must we be satisfied with the self-explanation effect or can and should we do more?

Previous work has attempted to recognize natural language explanations and then engage in a natural language dialogue with the student to refine and improve those explanations (e.g. [11]). And more recent work has attempted to field dialogue systems that incorporate more knowledge intensive automated recognition of students' elaborations during dialogue [4]. But so far, recognizing what the student meant is still very limited. And even if we step away from attempts at actual understanding, the performance for matching to canonical sets of answers is still relatively low (e.g. [5, 12]) compared to what can be achieved with short answer responses (e.g. [13]). Perhaps even more troubling is how sensitive students are to a system's failure to understand them [4]. Although a system can recover and move forward in a coherent manner, the students notice the lack of understanding. One possibility for this sensitivity may be that the errors are often quite different from those a human makes (e.g. the system fails to recognize a response as correct but a human clearly would).

Related work, which studied the impact of decisions about dialogue tactics [2], seems to have avoided some of these issues by substituting a human interpreter (wizard) for the automated interpreter. One goal of this substitution was to reduce the confounds of misunderstandings so that the system could focus on evaluating decision policies regarding whether to elicit or tell the explanations and justifications for statements made either by the system or the student. The human interpreter was presented with a list of canonical answers and was asked to find the best match for the student's response or to select "none of the above". There were significant differences in learning based just on varying decision policies about whether to elicit or tell the same content. This result suggests that being able to request explanations and justifications and being able to reduce the confounds of errors in matching to canonical answers has potential. But is there a practical way to include a human interpreter in a classroom setting? And how sensitive are students to problems that arise if their answer is close to correct but not a good match for any of the canonical answers?

First we will introduce the Rimac[1] system and its experimental setting and our approach for eliciting and assessing students' responses to requests for explanations/justifications. Next we will describe the data we have collected and provide a preliminary evaluation of the success of our approach for eliciting explanations/justifications. Finally, we will lay out our plans for exploring if there is value added to providing feedback on students' explanations.

2. THE RIMAC SYSTEM

Rimac is a natural-language intelligent tutoring system that engages students in dialogues that address physics concepts and principles, after they have solved quantitative physics problems. Much research has been devoted to identifying features of tutorial dialogue that can explain its effectiveness (e.g., [1]), so that these features can be simulated in natural-language tutoring systems. One hypothesis is that the highly interactive nature of tutoring itself promotes learning. Several studies indicate that our understanding of interactivity needs refinement because it cannot be defined simply by the amount of interaction nor the granularity of the interaction but must also take into consideration how well the interaction is carried out (e.g., [2]).

This need for refinement suggests that we should more closely examine the linguistic mechanisms evident in tutorial dialogue. Towards this end, we first identified which of a subset of co-constructed discourse relations correlate with learning and operationalized our findings with a set of nine decision rules which we implemented in Rimac [9]. To test for causality, we created two versions of Rimac: an experimental version that deliberately executes the nine decision rules within a Knowledge Construction Dialogue (KCD) framework, and a control KCD system that does not intentionally execute these rules. One of the rules in the experimental version of Rimac that is of particular relevance for this paper is: If the student gives no explanation for an answer to a complex question then ask for one.

Rimac was built using the TuTalk tutorial dialogue toolkit [6] but has been enhanced with additional dialogue features such as reformulation of student input (e.g., [7]). The dialogues are tutor-initiative only and are primarily short answer questions in order to keep the accuracy of automatic recognition high. However, the system does request student explanations at a few key points in the dialogues. It does not attempt automatic recognition of these responses but instead always follows-up with multiple choice answers for the explanation question and a request that the student select the best match for the explanation she just provided. In effect, we are asking the student to fulfill the role of a human interpreter for the system. We give the students brief verbal instructions for what is expected of them and why as we will explain in more detail below.

Figure 1 is an excerpt of an interaction between one of the students in our evaluation set and Rimac. This excerpt shows the system requesting an explanation at line 3. The student provides her response and then the system asks the student to select the best match for her response at line 5. The system then follows-up in lines 7-9 with formative feedback that is appropriate for what the student selected as the best match for her response.

1. T: We talked about the acceleration on the way up and on the way down, but what about at the very top of the jump? What was the direction of the acceleration at the top?
2. S: zero
3. T: Why do you think there wasn‘t any acceleration?
4. S: it had a zero velocity
5. T: Thanks! So what below is the best match for what you just entered? [menu choices: ‘Since the kangaroo is not moving, there isn‘t any acceleration | The kangaroo is changing direction, so it doesn‘t have an acceleration | I guessed | None of the above‘]
6. S: Since the kangaroo is not moving, there isn‘t any acceleration
7. T: There actually was acceleration at the top of the jump, even though the kangaroo wasn‘t moving.
8. T: The kangaroo wasn‘t moving for just a very brief moment and it was like that because its velocity was switching from upward to downward. To switch from an upward velocity to a downward velocity, the kangaroo‘s velocity must reach a point where it is 0 m/s.
9. T: Since acceleration represents the change in velocity and the velocity was changing in this situation, the kangaroo‘s acceleration must be non-zero.

Figure 1: Excerpt of actual student interaction that requests student's assistance with interpretation along with the resulting formative feedback

3. THE EXPERIMENTAL DESIGN

Students in five Pittsburgh area high schools interacted with one of the two versions of Rimac during two course units (kinematics and dynamics). They used the system for one to two class periods per unit. In this paper, we examine the dialogues from the kinematics unit only.

A day or two prior to using the system, students first took a pre-test, and then completed a homework assignment in which they solved four quantitative physics problems. In a subsequent class, they used the Rimac system and finally during the next class meeting took a post-test.

Just before students began using Rimac, we introduced them to the system and read the following to them regarding requests for explanations:

> "Sometimes it will ask you to explain your response. This is regardless of whether it thinks you were right or wrong.

[1] Rimac is the name of a river whose source is in the Andes. Its name is a Quechua word meaning "talking;" hence the nickname for Rimac: "talking river." We thus considered Rimac to be well suited to a dialogue system that we intended to embed in the Andes tutoring system.

> When it asks you to explain, please be sure to type in what you were thinking that lead you to your answer. You may have to think a bit about it. If you realize that you guessed or used your intuition, that's fine; just type that.
>
> It will then follow-up with a multiple choice question and ask you to pick what is the best match for what you just wrote. It is important that you pick the best match for the explanation you just wrote and not what looks like the best explanation. Rimac needs to know what your thought process was so it can do a better job of helping you understand the physics concepts involved in solving the problem.
>
> It asks you to do this matching for explanation questions because it cannot understand explanations accurately enough. However, for all the other answers you type in it is fairly accurate."

As the student and system begin the review of an assigned homework problem, Rimac first instructs her to view a brief video that describes how to solve the homework problem and then they engage in a reflective dialogue about that problem. See [10] for a more detailed description of the pilot study and planned analyses for testing the nine interactivity decision rules. Note that in this paper we are focusing on just one of these nine rules.

4. EVALUATING THE APPROACH FOR ELICITING EXPLANATIONS

We visited the high schools at two different points during their physics courses, once when the students were ready to learn the material covered in Rimac's kinematics problems and later when they were ready for Rimac's dynamics problems. The students who used the experimental version of Rimac, in which the interactivity rules are implemented, had the opportunity to discuss three kinematics problems and three dynamics problems. In this preliminary evaluation of our elicitation approach, due to time constraints, we have examined the explanation request data for the three kinematics problems only and for just a subset of the students who used this version of the system (93 of 193 students).

From this subset of the data, we extracted and reviewed the dialogue data for 16 of the 18 system turns where students were asked to interpret their own responses. Two of the 18 turns were requests to provide equations as justifications so we excluded those for this analysis. Although there are 93 students in this data sample, not all questions are asked of all students. The questions that are asked depend on the student's responses during the dialogue. In this sampling of students there were 251 instances of requests for explanations/justifications to examine. We reviewed each response and categorized the student's initial typed response as either relevant to the question asked, an acknowledgement that she had guessed or didn't know, or irrelevant to the physics domain. We listed this analysis per problem in order to see if there is any indication that the problem, the order in which the problems were discussed or the types of questions asked might influence the students' behaviors.

1. T: Why do you think that acceleration is upward?
2. S: nevermind
3. T: Which of the following is the best match for what you just entered? [menu choices: 'Because the kangaroo changed direction | Acceleration is always in the opposite direction as velocity | I guessed | None of the above']
4. S: Acceleration is always in the opposite direction as velocity

Figure 2: Example of an actual student reaction when asked to explain an incorrect answer

	Problem 1	Problem 2	Problem 3
relevant content	88% (123)	68% (50)	72% (28)
guessed/idk	5% (9)	15% (11)	23% (9)
irrelevant content	3% (7)	16% (12)	5% (2)

Table 1: A categorization of students' responses to requests for explanations

Table 1 shows that students rarely refused to respond (3-5%) for problems 1 and 3. Refusal to answer increased noticeably for problem 2 (16%) so it does appear that the questions asked for this problem may have impacted students' willingness to respond.

While most student replies that we counted as refusals to answer were simply replies of "because", we noted in problem 2 some instances of "ok" and "never mind" as in the excerpt shown in Figure 2. Note that in this case the system is asking the student to explain her incorrect answer. After saying "nevermind" the student may be selecting what she was thinking or something that looked like a true statement. Of the 16 system turns evaluated, 8 were in response to wrong answers. However, problem 2 accounted for 6 of those requests and problem 3 for the rest. The increased refusal to answer for problem 2 may be because there are more requests to explain incorrect responses. When students self-explain and recognize they have made an error, it appears they sometimes try to figure out how to retract their answer or evade the expected follow-up. However, if the student does not realize her error, it does seem that the formative feedback could be beneficial. For example, consider the dialogue excerpt shown in Figure 3 where the student is asked to explain a wrong answer in line 6 and the formative feedback is provided in lines 8-13.

Next we evaluated how well the students performed in their role as human interpreter. We reviewed each response by following the training given to human interpreters for the experiments described in [2] which used the Cordillera system (i.e. how would I match this response to the choices provided). One of the reviewers of the students' responses (the main author) was also one of the human interpreters (wizard) for the first two versions of the Cordillera system. So it could be instructive to informally compare and contrast the interpretation tasks for the two systems and populations from the perspective of a wizard.

1. T: If the stone is thrown with a larger horizontal initial velocity, what happens to the time spent in the air (i.e. will it be the same or larger?)
2. S: larger
3. T: Hmm...So why did you think this?
4. S: more time
5. T: Thanks for explaining your reasoning.
6. T: So which of the reasons below best matches your explanation above for why you think the stone will stay longer in the air? [menu choices: ‘I used my intuition but I am not sure how to explain it in physics terms | Throwing it harder means the stone will not fall right away but will continue horizontally for a longer time before beginning to fall | Throwing it harder makes the stone spin and float on the air better | I‘ve seen this happen before | Because it is going farther, it will take longer to get there | Another reason‘]
7. S: Because it is going farther, it will take longer to get there
8. T: That doesn‘t sound right to me. What equation would you use to reason that out? [menu choices: ‘$d_x = v_x*t|d_x = v_x*t+0.5*a_x*t^2|vf_x^2 = vi_x^2+2*a_x*d_x$‘]
9. S: $d_x = v_x * t + 0.5 * a_x * t^2$
10. T: Almost. What is the value of a_x?
11. S: 0 m/s^2
12. T: Right, so then the equation becomes $d_x = v_x * t$.
13. T: The horizontal component of the motion is constant velocity (not accelerating), so $d_x = v_x * t$ applies. In this case, because the velocity is bigger, the displacement will be bigger, but the time will not be.

Figure 3: Excerpt of actual student interaction where system requests explanation of wrong response

The Cordillera students were all undergraduates and their explanations were longer and required more effort to interpret and match. However, there was usually one clear candidate for the match and when matching to a correct response the criteria were that the necessary and sufficient details were present or could be easily inferred and no additional details signalled an error in thinking. The choices were authored to provide the minimum that would be needed to qualify as a complete answer. While wizards did not have to be physics experts, they did need to understand the physics concepts being discussed.

Context: Problem solved for homework “A red colored stone is thrown horizontally at a velocity of 5.0 m/s from the roof of a 35.0 m building and later hits the ground below. What is the red stone’s horizontal displacement? Ignore the effects of air friction.”
Question: Why did we need to find the time first?
Choices:

1. time is the same in both directions
2. d = vt
3. we don’t have enough information to solve for displacement in the horizontal direction
4. we can find the displacement if we know how long it is moving at the given velocity
5. another reason

Figure 4: An example of where some choices offered to students for matching are related to the same underlying explanation (as in choices 1,3 and 4)

In contrast, the Rimac students were all in high school and their explanations were relatively short. We did not expect students to do well with a set of minimal match choices since we assume you need to understand the physics concepts to determine whether an answer actually matches. So instead the Rimac dialogue authors provided responses for matching that were intended to be closer to what a student might say and were based on input from teachers and responses collected during pilot testing. As a result some of the choices offered to students for matching varied only in the detail provided or how it was expressed. But these similar choices present the same formative feedback when selected. For example, in Figure 4, choice 2 is close to a good explanation but requires more detail to be complete while choices 1,3 and 4 are all related to the same underlying explanation. If the student selects 1,3 or 4 as a match then the underlying explanation is presented as an acknowledgement and may be interpreted by the student as a reformulation. If the student selects choice 2 then the system provides scaffolding that elicits the missing details.

So during our review of students’ response matching, we selected all that we considered to be potential matches and not just the best match. The rationale was that if a student selected one of a similar set of responses that had details that were missing in her response, a wizard cannot know whether the student’s self-explanation included these details and she chose not to express them or whether she thought more detail was necessary and was trying to avoid formative feedback.

After reviewing the student responses we counted the number of times we disagreed with their match choices. Again we present the results per problem. Table 2 shows that students’ performance may be similar to that of an automated explanation matcher. The larger disagreement for problem 2 could be due to students possibly trying to evade further feedback when they were asked to explain an incorrect answer or could be related to the questions or answer choices offered. If deserves a closer look in future work to see if a reason can be identified.

However, overall the students seem less perturbed by the results of their matching behaviors. They still continued to respond to the requests for explanations as shown by the

	Problem 1	Problem2	Problem 3
agree	78% (108)	59% (43)	74% (29)
disagree	22% (31)	41% (30)	25% (10)

Table 2: Reviewer agreement with students' matches of their responses

small increase in irrelevant content in Table 1, which remains low with an increase from 3 to 5% when moving from the first to last problem. The increase from problem 1 to problem 3 in "guessed/idk" could be due to fatigue, the explanations requested or more specifically asking for more explanations for incorrect answers in problems 2 and 3. Although the number of "guessed/idk" decreased from problem 2 (11) to problem 3 (9), recall that some students completed problems in two class sessions and some in one. This was because of differences in the length of classes across schools.

To give an idea of an upper bound for agreement, we do not expect 100% agreement between the reviewer and a trained human interpreter (wizard). When offline reviewers examined the selection choices made by the real-time human interpreters for the Cordillera system for just the most difficult student responses (i.e. those that fell into the "none of the above" category), the reviewer disagreed with 1% of the assignments to this category [8]. However, the lower bound that is allowable for matching when students are acting as the interpreter is still an open question. It will depend on whether formative feedback on the explanation related to their match choice is beneficial.

By the time of the workshop, we expect to have completed the above analyses for all students for the kinematics problems.

5. PLANS FOR EVALUATING THE FORMATIVE FEEDBACK GIVEN ON EXPLANATIONS

Recall that in the instructions we read to students we asked that they match the response they gave rather than picking what looks like the best response. We offer motivation to do this by pointing out that the system needs to know their thought processes so that it can provide better help for them. We are assuming that the formative feedback of a good match will be better than the "none of the above" feedback. However, this remains to be seen.

But because our experiment was not testing this specific hypothesis, we cannot answer this question directly (e.g. compare to a condition in which the formative feedback is always the "none of the above" feedback). However, we can test for correlations between various match qualities (i.e. trained reviewer agreed or disagreed with student) and learning of the concepts addressed by the requested explanation. This would suggest how important it is for students to receive more adapted formative feedback. In addition, we can test for gains on concepts covered in an explanation when the student's explanation is incorrect and relative to the quality of the match the student provided. This could suggest whether the feedback that followed was beneficial.

This preliminary analysis of the effects of formative feedback is forthcoming. We are currently scoring the pre and post-tests, which (when completed) will allow us to measure learning of particular concepts.

6. ACKNOWLEDGEMENTS

This research is supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A10063 to the University of Pittsburgh. The opinions expressed are those of the authors and do not necessarily represent the views of the Institute or the U.S. Department of Education.

7. REFERENCES

[1] B. Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13:4–16, 1984.

[2] M. Chi, K. VanLehn, D. Litman, and P. Jordan. An evaluation of pedagogical tutorial tactics for a natural language tutoring system: A reinforcement learning approach. *International Journal of Artificial Intelligence in Education*, 21:83–113, 2011.

[3] M. T. Chi, N. De Leeuw, M.-H. Chiu, and C. LaVancher. Eliciting self-explanations improves understanding. *Cognitive science*, 18(3):439–477, 1994.

[4] M. O. Dzikovska, P. Bell, A. Isard, and J. D. Moore. Evaluating language understanding accuracy with respect to objective outcomes in a dialogue system. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 471–481. Association for Computational Linguistics, 2012.

[5] A. C. Graesser, P. Wiemer-Hastings, K. Wiemer-Hastings, D. Harter, N. Person, and the TRG. Using latent semantic analysis to evaluate the contributions of students in autotutor. *Interactive Learning Environments*, 8:129–148, 2000.

[6] P. Jordan, B. Hall, M. Ringenberg, Y. Cui, and C. Rosé. Tools for authoring a dialogue agent that participates in learning studies. In *Proceeding of Artificial Intelligence in Education Conference*, pages 43–50, 2007.

[7] P. Jordan, S. Katz, P. Albacete, M. Ford, and C. Wilson. Reformulating student contributions in tutorial dialogue. In *Proceedings of 7th International Natural Language Generation Conference*, pages 95–99, 2012.

[8] P. Jordan, D. Litman, M. Lipschultz, and J. Drummond. Evidence of misunderstandings in tutorial dialogue and their impact on learning. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education (AIED), Brighton, UK, July*, 2009.

[9] S. Katz and P. Albacete. A tutoring system that simulates the highly interactive nature of human tutoring. *Educational Psychology (Special Issue on Advanced Learning Technologies)*, in press.

[10] S. Katz, P. Albacete, M. Ford, P. Jordan, M. Lipschultz, D. Litman, S. Silliman, and C. Wilson. Pilot test of a natural-language tutoring system for physics that simulates the highly interactive nature of human tutoring. In K. Yacef and H. Lane, editors,

Proceedings of Artificial Intelligence in Education Conference, 2012.

[11] M. Makatchev and K. VanLehn. Analyzing completeness and correctness of utterances using an atms. In *Proceeding of the 2005 conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology*, pages 403–410, 2005.

[12] V. Rus and A. C. Graesser. Deeper natural language processing for evaluating student answers in intelligent tutoring systems. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 1495. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[13] S. Siler, C. P. Rosé, T. Frost, K. Vanlehn, and P. Koehler. Evaluating knowledge construction dialogs (kcds) versus minilessons within andes2 and alone. In *Workshop W6 on Empirical Methods for Tutorial Dialogue Systems*, page 9, 2002.

Must Feedback Disrupt Presence in Serious Games?

Matthew Jensen Hays, H. Chad Lane, Daniel M. Auerbach
University of Southern California
Institute for Creative Technologies
12015 E Waterfront Dr
Playa Vista CA 90094 USA
310-448-5398
{hays,lane,auerbach}@ict.usc.edu

ABSTRACT

Serious games are generally designed with two goals in mind: promoting learning and creating compelling and engaging experiences (sometimes termed *a sense of presence*). Presence itself is believed to promote learning, but serious games often attempt to further increase pedagogical value. One way to do so is to use an intelligent tutoring system (ITS) to provide feedback during gameplay. Some researchers have expressed concern that, because feedback from an ITS is often *extrinsic* (i.e., it operates outside of the primary game mechanic), attending to it disrupts players' sense of presence. As a result, learning may be unintentionally *hindered* by an ITS. However, the most beneficial conditions of instruction are often counterintuitive; in this paper, we challenge the assumption that feedback during learning hinders sense of presence. Across three experiments, we examined how an ITS that provided extrinsic feedback during a serious game affected presence. Across different modalities and conditions, we found that feedback and other ITS features do not always affect presence. Our results suggest that it is possible to provide extrinsic feedback in a serious game without detracting from the immersive power of the game itself.

Keywords

presence, immersion, learning, feedback, serious games, tutoring

1. WHAT'S IN A GAME?

We have all had the experience of being engrossed in an artificial experience, whether it's a good book, an epic movie, a round of golf, or a couple levels of *Angry Birds* on a long elevator ride. Several features of games, especially, can make hours fly by, unnoticed. The interactivity of games draws players' attention from non-game thoughts and stimuli. The rules of the game, too, are designed to add uncertainty and difficulty—and eventual reward—to the pursuit of an objective. Putting a ball into a cup is made fun, for example, by requiring that one use golf clubs to do so—rather than simply picking up the ball, walking over to the cup, and dropping it in. The eventual reward (sinking a putt) compels players to persist and eventually improve.

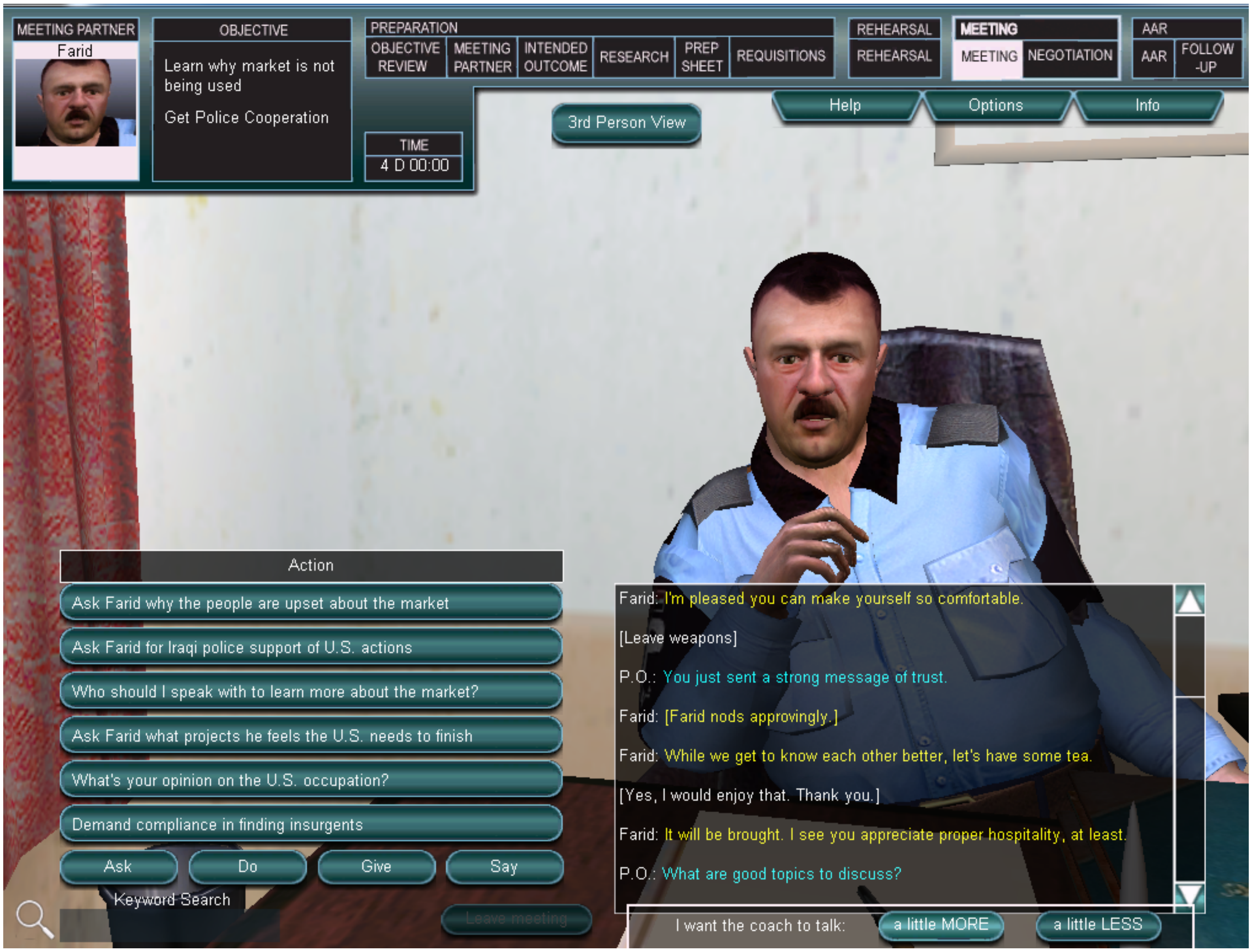
Real-world games are fun, in part, because they take place in an environment that supports continued play (e.g., a golf course). Digital games, instead, must transport a player to the world of the game. This experience of being in the world of the game is sometimes referred to as a sense of *presence* [1]. Presence can be measured in several ways. The Temple Presence Inventory (TPI), for example, is a robust instrument for estimating the feeling of non-mediation in a multimedia experience [2]. The TPI consists of a series of statements to which participants respond to items such as "How often did you want to or did you make eye contact with a person you saw/heard?" with ratings between 1 (never) and 7 (always). These statements are organized into several subscales, which correspond to various aspects of the experience that contribute to the sense of non-mediation. The two subscales we used were *social* (the experience of direct interaction with an artificial counterpart) and *spatial* (the experience of direct contact with an artificial environment).

2. WHAT'S IN A SERIOUS GAME?

In addition to the standard traits of a digital game (e.g., the difficult pursuit of an in-game objective, creating a sense of presence), *serious games* feature an objective outside the game itself. By "playing" a serious game, one becomes better at a real-world task—or is at least better prepared to learn that task from subsequent instruction or practice [3]. Examples of serious games include CyberCIEGE, which is designed to teach people about the functions of computer network security measures. Another example is Spent, a simple simulation of a U.S. Citizen's experience at the poverty line in a difficult economy with no bootstraps on which to pull. The difficulty, interactivity, and reward structure of serious gameplay can compel students to persist in learning something they would otherwise find dry or boring.

Serious games are also used in part because the sense of presence created by gameplay may improve learning [4, but see 5, 6, 7]. On the other hand, the outside-the-game objective may be in conflict with that intent. Of course, a game-player's sense of presence in a serious (or otherwise overtly educational) game may be disrupted by poorly integrated pedagogical content. For example, some educational games alternate between play and instruction. But even well integrated instructional content may be distracting; the user may occasionally stop to consider how to apply what they are learning to similar real-world tasks. If presence affects learning, this withdrawal may be detrimental.

This potential conflict may be exacerbated when features that are intended to facilitate training are added to a serious game. These


MEETING PARTNER
Farid
OBJECTIVE
PREPARATION
OBJECTIVE REVIEW
MEETING PARTNER
INTENDED OUTCOME
RESEARCH
PREP SHEET
REQUISITIONS
REHEARSAL
REHEARSAL
MEETING
MEETING
NEGOTIATION
AAR
AAR
FOLLOW -UP
Help
Options
Info
3rd Person View
TIME
4 D 00:00
Action
Ask Farid why the people are upset about the market
Ask Farid for Iraqi police support of U.S. actions
Who should I speak with to learn more about the market?
Ask Farid what projects he feels the U.S. needs to finish
What's your opinion on the U.S. occupation?
Demand compliance in finding insurgents
Ask
Do
Give
Say
Keyword Search
Leave meeting
I want the coach to talk:
a little MORE
a little LESS

Learn why market is not being used

Get Police Cooperation

Farid: I'm pleased you can make yourself so comfortable.

[Leave weapons]

P.O.: You just sent a strong message of trust.

Farid: [Farid nods approvingly.]

Farid: While we get to know each other better, let's have some tea.

[Yes, I would enjoy that. Thank you.]

Farid: It will be brought. I see you appreciate proper hospitality, at least.

P.O.: What are good topics to discuss?

Figure 1. A meeting in BiLAT. In the transcript pane (bottom right), the feedback from the ITS-driven coach appears as blue text. Below that are buttons used to adjust how frequently the coach (P. O., above) decides to intervene (Experiments 2 and 3).

features may directly interfere, or may simply underscore that the player is *using* the game to achieve the external goal, as opposed to *playing* the game because it is fun.

One such feature is an intelligent tutoring system (ITS). An ITS is a computer program or computing device that factors student performance into when and how it generates and provides guidance [8]. The development of ITSs (and other learning-centric game features) is usually guided by principles of cognitive psychology and instructional design [8-10]. However, those principles are often developed in experimental laboratories, in which motivation and fun may not be priorities. Thus, ITSs may provide pedagogically valid feedback, but they may do so in a way that further deepens the rift between gameplay and learning. The goal of the studies reported in this paper was to determine whether extrinsic feedback from an ITS necessarily negatively affects learners' sense of presence when playing a serious game.

3. BILAT: A SERIOUS GAME ABOUT CROSS-CULTURAL NEGOTIATION

The serious game we chose to use for our investigation is the Enhanced Learning Environments with Creative Technologies for Bilateral negotiations (ELECT BiLAT), a screenshot from which is shown in Figure 1. BiLAT provides an environment in which learners can prepare for, execute, and review cross-cultural meetings with virtual characters. The instructional design and underlying structure are focused on knowledge components that relate to culture and negotiation skills.

Before a meeting, players research their meeting partner, learning about his/her interests and experiences. This research provides information that can help the character establish a personal connection with the character during their meeting. Once the meeting begins (shown in Figure 1), players interact with the characters by selecting an action from a menu system of pre-authored actions (e.g., Ask "Who should I speak with to learn more about the market?"). The character responds to the learner with a synthesized voice and physical gestures. The player and the virtual character thus conduct a turn-based interaction, and the transcript of the meeting appears on screen in the panel at the bottom right of Figure 1.

Although dozens of variables govern the actions of the character and the responses that will be chosen, the variable of primary importance is trust. BiLAT characters display a variety of emotions in their responses, but trust is the persistent record of how well players have used their interpersonal and intercultural

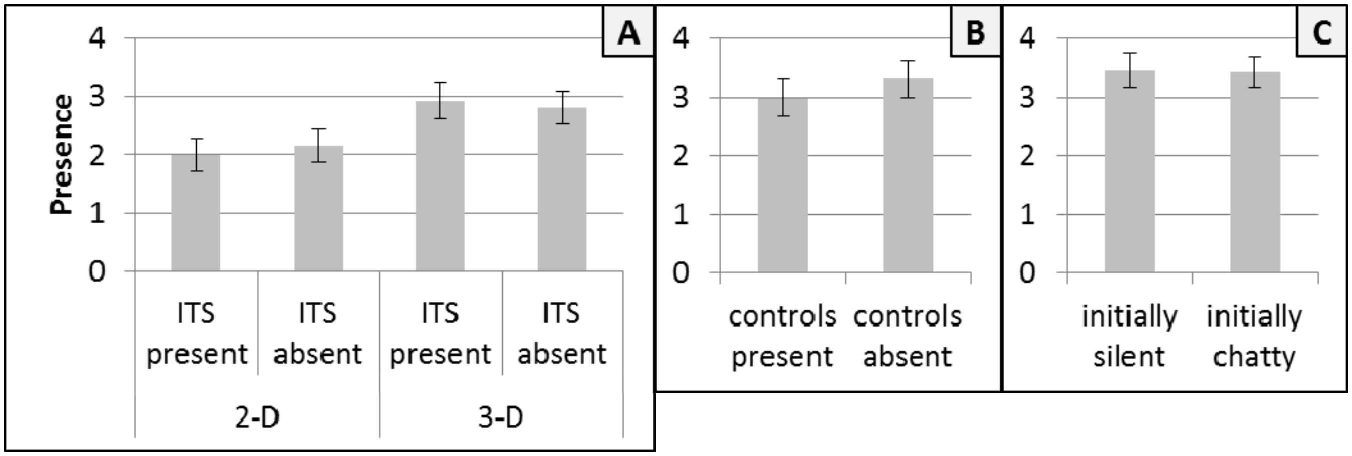


Figure 2. Results from all three experiments. Panel A displays presence as a function of interface richness and ITS activation in Experiment 1. Panel B displays presence as a function of ITS interactivity in Experiment 2. Panel C displays presence as a function of initial ITS feedback frequency in Experiment 3. Error bars represent the standard error of the mean.

skills. In the simulation, trust is a major factor in whether BiLAT characters will agree to negotiate and what deals they will accept. A mistrusting character may demand unfair deals or refuse to negotiate. (For a more detailed description of BiLAT's development and functionality, please see [11, 12].)

The characters' responses and decisions can be considered *internal feedback*. They help the player grasp the knowledge components through the primary interaction that constitutes gameplay. For example, if the player decides to offer the character a bottle of wine as a gift, the character will be offended and say so: "I can't believe you'd even bring that into my home." Depending on what the player has encountered both in and out of BiLAT, the player may conclude that the character does not like wine or that wine is a culturally inappropriate gift.

During BiLAT gameplay, learners can be assisted by an ITS. In meetings with characters, the ITS takes the form of a disembodied, omniscient "coach." The player can read the coach's input in the transcript pane, but the meeting partner is not aware of the coach's presence or input. In other words, the coach is an angel on the player's shoulder. The input the coach provides is outside of the primary interaction that constitutes gameplay; it is *external feedback*.

The coach can provide guidance about past actions ("A bottle of wine probably wasn't the best gift.") or hints about future actions ("What gift can you give Hassan as a gesture of goodwill?"). This advice can be either very general (i.e., focused on the underlying knowledge components) or very specific to something a player has done. For example, the coach could decide to say "Don't give Hassan a bottle of wine" or "Make sure your gifts are culturally appropriate." (For a detailed description of the ITS architecture, please see [13].)

4. EXPERIMENT 1: THE EFFECTS OF EXTERNAL FEEDBACK ON PRESENCE

In Experiment 1, we examined the effects of explicit ITS feedback on learners' sense of presence during BiLAT gameplay. The manipulation was straightforward: whether the ITS was active or inactive during gameplay. We also added another manipulation: whether the sensory experience was rich or poor. Our goal in adding this manipulation was to ensure that we would be able to detect effects on presence with our system, procedure, and participation numbers. Thus, one group of the participants encountered the standard BiLAT experience: a 3-D environment in which a virtual character with realistic body language talks to the player in accented English. The other group of participants encountered a simplified, silent, primarily text-based 2-D environment. We held constant all other aspects of the system for the two groups. Specifically, the BiLAT characters drew from the same sets of utterances and the coach used the same algorithms to decide when to intervene. Only the interface of the two groups' experiences differed. After interacting with the system in one of the four resultant (randomly assigned) conditions, the participants completed the TPI.

Panel A of Figure 2 shows that there was a main effect of interface on presence. A greater sense of presence was created by the 3-D interface (M = 2.88, SE = .21) than by the 2-D interface (M = 2.08, SE = .20): $F(1, 45) = 7.86$, p = .007. There was not a main effect of ITS activation on presence. Indeed, presence ratings were similar in the active-ITS condition (M = 2.46, SE = .20) and the inactive-ITS condition (M = 2.49, SE = .20): $F < 1$, *ns*. There was also no interaction between interface and ITS activation on presence: $F < 1$, *ns*. It appears that receiving extrinsic feedback from an ITS does not necessarily affect presence. Thus, any pedagogical benefit provided by the ITS appears not to burden the immersive experience.

5. EXPERIMENT 2: THE EFFECTS OF FEEDBACK CONTROLS ON PRESENCE

In Experiment 1, the activity of the ITS was entirely out of the participants' control. In Experiment 2, we added interactivity to the ITS. We gave the participants the ability to modify the coach's behavior. We thought that this interactivity might cause the participants to attend to the coach (or the external training goal of the serious game) in a way that would disrupt presence.

There were two groups of participants, both of which encountered the standard, 3-D BiLAT system with the coach operating according to its default algorithms. One of the groups was also provided with "coach controls." These controls took the form of the buttons seen in the bottom right corner of Figure 1. These buttons suggested to the participants that they could nudge (up or down) the frequency with which the coach decided to intervene.

The controls, however, were only cosmetic (although they still visually and aurally behaved like other in-game buttons). We chose to display but disable them in order to manipulate the participants' *belief* that they could control the coach without allowing learning, performance, success, or frustration to vary uncontrollably. After interacting with the system in one of the two (randomly assigned) conditions, the participants completed the TPI.

Panel B of Figure 2 shows that there was no main effect of ITS controls on presence: $F(1, 22) < 1$, *ns*. This result provides more evidence that even direct interaction with an ITS outside the primary game mechanic does not necessarily disrupt presence.

6. EXPERIMENT 3: THE EFFECT OF ITS HELPFULNESS ON PRESENCE

Experiment 3 was designed to extend Experiment 2. Our goal was to determine whether the BiLAT ITS could deliver feedback in a way that would disrupt presence. To that end, we modified the coach's feedback-timing algorithms to draw even more attention to the ITS than in Experiment 2. For one group of participants, the coach began the session in complete silence. For the other group of participants, the coach began the session by speaking up on every single turn. We activated the "nudge" controls, which were merely cosmetic in Experiment 2, to encourage the participants to interact with the ITS as much as possible. Each press of "a little more" or "a little less" changed (by 5%) the probability that the coach would speak up on the next turn. After interacting with the system in one of the two (randomly assigned) conditions, the participants completed the TPI.

As can be seen in Panel C of Figure 2, the participants in both conditions provided similar presence ratings: $F(1, 22) < 1$, *ns*. That is, whether the participants' experience began with constant chatter or complete silence from the ITS, their sense of presence remained relatively unaffected. Moreover, in comparing the three panels in Figure 2, it is clear that the participants' overall ratings were similar across all three experiments—despite drastic differences in feedback algorithms and ITS interactivity. It seems that, unless an ITS is designed with the express purpose of disrupting gameplay, it may not interfere with the immersion created by a serious game.

7. GENERAL DISCUSSION

Interpersonal and intercultural skills, to be frank, may not be the most compelling instructional topics. However, when playing BiLAT, players and participants become very engaged. A participant in one study, when meeting with a particularly stubborn character, took off his headphones and threw them across the room, saying "I *know* he wants to agree to it, and he's just trying to give me a headache!"

Our research demonstrates that this sense of presence is not necessarily disrupted when external feedback from an ITS is added to a serious game. Further, learners can even be instructed to directly interact with the ITS, yet still suffer no decrement to self-reported presence. On the other hand, the use of a single, self-report measure of presence is a limitation of the present study. A more compelling case may be presented by including corroborating physiological data. (We did not examine measures of performance or learning because it would have been impossible to disentangle from each other the effects of feedback on presence, feedback on learning, and presence on learning.)

Although these results may seem surprising, external stimuli interrupt engaging experiences quite frequently, often with no negative results. Many people have put down and then resumed an engrossing book—and been able to reinstate their enjoyment of and engagement with the story. Perhaps a compelling narrative or rewarding gameplay may make some serious and educational games robust to interruptions, as well. In these cases, people may be able to suspend and resume their engagement as they wish. If so, it is interesting to consider the extent to which developers can add pedagogically focused game features without sacrificing learners' immersion. It is reasonable to assume there is some limit to the intrusiveness an ITS can exhibit while still being effective—but the present studies suggest that that limit is above zero.

8. ACKNOWLEDGMENTS

The work depicted here was sponsored by the U.S. Army. Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

9. REFERENCES

[1] Lombard, M. and Ditton, T. *At the heart of it all: The concept of presence.*, 1997.

[2] Lombard, M., Ditton, T. and Weinstein, L. *Measuring presence: The Temple Presence Inventory*. 2009.

[3] Dede, C. Immersive interfaces for engagement and learning. *Science*, 323, 2009, 66-69.

[4] Rowe, J. P., Shores, L. R., Mott, B. W. and Lester, J. C. Integrating learning, problem solving, and engagement in narrative-centered learning environments. *International Journal of Artificial Intelligence in Education*, in press.

[5] McQuiggan, S., Rowe, J., Lee, S. and Lester, J. *Story-based learning: The impact of narrative on learning experiences and outcomes*. 2008.

[6] Lane, H. C., Hays, M. J., Auerbach, D. and Core, M. *Investigating the relationship between presence and learning in a serious game*. 2010.

[7] Moreno, R. and Mayer, R. E. Personalized messages that promote science learning in virtual environments. *Journal of Educational Psychology*, 96, 2004, 165-173.

[8] Woolf, B. *Building intelligent interactive tutors*. Morgan Kaufmann, 2008.

[9] Mayer, R. E. *Multimedia Learning*. Cambridge University Press, 2001.

[10] Shute, V. J. and Psotka, J. *Intelligent tutoring systems: Past, present, and future*. MacMillan, 1996.

[11] Hill, R. W., Belanich, J., Lane, H. C., Core, M., Dixon, M., Forbell, E., Kim, J. and Hart, J. *Pedagogically structured game-based training: Development of the Elect BiLAT simulation*. 2006.

[12] Kim, J., Hill, R. W., Durlach, P. J., Lane, H. C., Forbell, E., Core, M., Marsella, S., Pynadath, D. V. and Hart, J. BiLAT: A game-based environment for practicing negotiation in a cultural context. *International Journal of Artificial Intelligence in Education*, 2009.

[13] Lane, H. C., Hays, M. J., Auerbach, D., Core, M., Gomboc, D., Forbell, E. and Rosenberg, M. Coaching intercultural communication in a serious game. *Proceedings of the 18th International Conference on Computers in Education*, 2008, 35-42.

Individual differences in the effect of feedback on children's change in analogical reasoning

Claire E. Stevenson
Leiden University, Psychology Methods & Statistics
Wassenaarseweg 52, Postbus 9555
2300 RB Leiden, The Netherlands
+31 71 527 3789
cstevenson@fsw.leidenuniv.nl

Paul A. L. de Boeck
Ohio State University, Dept. of Psychology
232 Lazenby Hall, 1827 Neal Ave,
Columbus, OH, 43210, USA
+1 614 292 4131
deboeck.2@osu.edu

Wilma C. M. Resing
Leiden University, Developmental Psychology
Wassenaarseweg 52, Postbus 9555
2300 RB Leiden, The Netherlands
+31 71 527 3789
resing@fsw.leidenuniv.nl

Willem J. Heiser
Leiden University, Psychology Methods & Statistics
Wassenaarseweg 52, Postbus 9555
2300 RB Leiden, The Netherlands
+31 71 527 3789
heiser@fsw.leidenuniv.nl

ABSTRACT

Various forms of feedback are used in formative assessment and interactive learning environments. The effects of different types of feedback are often examined at a group level. However, effective feedback may differ in learners with different characteristics or between learners at different stages in the learning process. In this paper explanatory item response theory (IRT) models are used to examine individual differences in feedback effects in children's performance on a computerized pretest-training-posttest assessment of analogical reasoning. The role of working memory and strategy-use as well as interactions between these factors were examined in a sample of 1000 children who received either stepwise elaborated feedback, repeated simple feedback or no feedback during the training sessions. The results show that working memory efficiency significantly predicted initial ability and confirm that elaborate feedback is the most effective form of training in this particular interactive learning environment. Furthermore, children with initially less advanced strategy-use benefitted far more from each type of feedback than the children displaying more advanced strategies and this was unrelated to working memory efficiency. In children with advanced strategy-use working memory appears to moderate the effect of training. Explanatory IRT analyses appear useful in disentangling the effects of learner characteristics on performance and change during formative assessment and could possibly be used in optimizing feedback in computerized training and assessment environments.

Keywords

Figural analogies, measuring change, item response theory, formative assessment

1. INTRODUCTION

Computer-based interactive learning environments have enormous potential in optimizing learning by providing feedback tailored to an individual's instructional needs. However, determining what type of feedback best optimizes the learning of a particular task for a particular individual is a complex endeavor. The effectiveness of different types of feedback is not always clear-cut. Furthermore, individual differences may be present in how effective each of these types of feedback is at different stages in the learning process.

In formative assessment different types of feedback can be used. Shute distinguished a range of feedback-types from simple forms such as verification of correct response to elaborated feedback where errors may be flagged, an opportunity to try again is provided and/or strategic prompts are given on how to proceed with the problem [Shutte 2008]. Kluger and DeNisi [1996] argued that although simple feedback, such as information on correctness of response or provision of the correct answer, has the reputation of improving performance on tasks, its effect is not clear-cut and only improves performance or learning in two-thirds of the studies included in their meta-analysis. Furthermore, more recent research demonstrates that elaborate feedback, such as providing scaffolds or an explanation, is generally more effective than simple outcome feedback [Hattie and Gan, 2011; Narciss and Huth 2006; Shutte 2008]. For example, a meta-analysis of effects of different forms of item-based feedback in computer-based environments reports that elaborated feedback shows higher effect sizes than simple outcome feedback, especially in higher-level learning outcomes, where transfer of previous learning to new situations or tasks is required [van der Kleij et al. 2013].

In the case of formative assessment the aim is to optimize learning at an individual level. In this educational setting the assumption is that there are individual differences both in initial ability as well as the effect of different types of feedback during an individual's learning process. Furthermore, different types of feedback may be more effective during successive stages in the learning process. However, effective feedback may differ for different types of learners or at different stages in the learning process. For example working memory efficiency and strategy-use have been implicated as predictors of performance in (computer-based) learning [Siegler and Svetina, 2002; Stevenson 2012; Tunteler et al. 2008]. In this study these factors were examined in conjunction with feedback-type as possible predictors of learning outcomes in a computerized training and assessment of analogical reasoning.

Initial ability or learning stage especially appears to play an important role in the effect of different forms of feedback on learning [Hattie and Timperley 2007]. For example, in a previous study on children's change in analogical reasoning training utilizing repeated simple feedback was contrasted with graduated prompting techniques, a form of elaborated feedback where increasingly specific strategic hints guide the child to the correct solution [Campione and Brown 1987; Resing and Elliott, 2011]. The researchers found that although graduated prompts led to greater performance gains on the whole, this form of training was most effective for children who performed poorly on the pretest [Stevenson et al. 2013a]. These results could not be explained by ceiling effects or regression to the mean. Furthermore, this result coincided with other cognitive training studies in various domains where interventions were generally more effective in initially lower performing or at-risk populations. Does this mean that providing elaborate versus simple feedback is not necessarily beneficial for more advanced learners?

To further explore the role of initial ability on feedback effects we examined the role of children's initial solution strategies (analogical versus non-analogical, see Figure 1) in the effect of three types of feedback: (1) step-wise elaborated feedback, (2) repeated simple feedback or (3) no feedback. The hypothesis was that children with initially weaker analogical reasoning strategies, characterized by "duplicate" (copying object next to empty box) solutions or "other / creating a zoo" solutions would benefit most from more elaborate forms of feedback whereas children who were already capable of applying analogical reasoning strategies (providing (partially) correct solutions) would not show differential benefit in the different types of feedback training. The role of working memory, which has often been shown to be related to analogy solving skills, but not always able to account for children's change in analogical reasoning [Stevenson et al. 2013b], was also taken into account in these analyses.

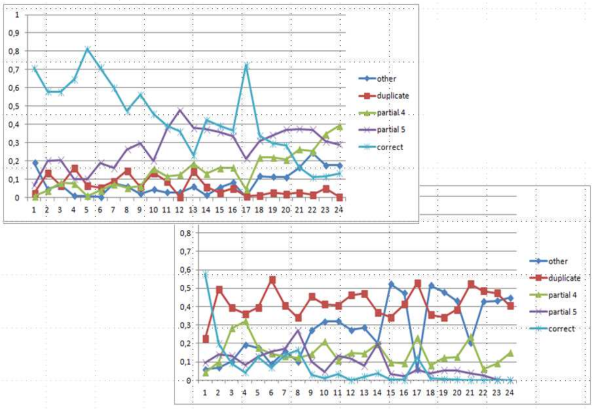


Figure 1. Depiction of strategy distribution within two pretest strategy groups: non-analogical reasoners (top left) and analogical reasoners (bottom right).

2. METHODS

2.1 Sample

1000 children from five age-groups (kindergarten, first through fourth grade) were recruited from public elementary schools of similar middle class SES in the south-west of the Netherlands. The sample consisted of 374 boys and 626 girls, with a mean age of 7 years, 3 months (range 4.9-11.3 years). The schools were selected based upon their willingness to participate and written informed consent for children's participation was obtained from the parents.

2.2 Design & Procedure

The data utilized in this study is a combination from five separate studies utilizing a pretest-intervention-posttest control-group design [Stevenson 2012]. In each study the children were randomly blocked to the step-wise elaborative feedback (graduated prompts), repeated simple feedback or a control condition without feedback based on their scores on a cognitive ability reasoning subtest (visual exclusion from the Revised Amsterdam Children's Intelligence Test [Bleichrodt et al. 1987] or the Standard Progressive Matrices [Raven et al. 2004]). The three intervention conditions presented in this study are: (1) stepwise elaborate feedback, (2) repeated simple feedback, or (3) no feedback. Four analogy testing and intervention sessions took place weekly and lasted 20-30 minutes each. Prior to the analogy testing sessions the children were also administered the Automated Working Memory Assessment to assess verbal (subtest listening recall) and visuo-spatial (spatial span) working memory [Alloway 2007]. All participants were tested individually in a quiet room at the child's school by educational psychology students trained in the procedure.

2.3 Analogical reasoning assessment

AnimaLogica was used to test and train children in analogical reasoning [Stevenson 2012]. The figural analogies (A:B::C:?) comprise of 2x2 matrices with familiar animals as objects (see Figure 2). The animals changed horizontally or vertically by color, orientation, size, position, quantity or animal type. The number of transformations – or object changes – provide an indication of item difficulty [Mulholland et al. 1980]. The children were asked to construct the solution to the analogy using drag & drop functions to place animal figures into the empty box in the lower left or right quadrant of the matrix. A maximum of two animals were present in each analogy. These were available in three colors (red, yellow, blue) and two sizes (large, small). The orientation (facing left or right) could be changed by clicking the animal figure. Quantity was specified by the number of animal figures placed in the empty box. Position was specified by location of the figure placed in the box.

The pretest and posttest items were isomorphs [Freund and Holling 2011] in which the items only differ in color and type of animal, but utilize the exact same transformations to ensure the same difficulty level. The number of items different per age group but included overlapping items ability could be estimated reliably using item response models. The internal consistency of each of the versions was considered very good with α ≥ .90.

Before each testing or training session two example items were provided with simple instructions on how to solve the analogies. If the child's solution was incorrect the correct solution was shown before proceeding to the next item. During the testing phases the remaining items were administered without feedback.

Table 1.

Overview of the prompts used in the elaborative feedback condition.

Prompt	Verbal Instruction
0	Here's a puzzle with animal pictures. The animals from this box have been taken away. Can you figure out which ones go in the empty box?
1	Do you remember what to do? Look carefully. Think hard. Now try to solve the puzzle.
2	This animal picture changes to this one. This one should change the same way.
3	So what changes here? Ok remember this one changes the same way.
4	See, this picture changes to this one because…
5	Which animal goes in the empty box? The elephant or the horse? What color should it be? Red, Yellow or Blue? …Size? Quantity? Orientation? Position?...

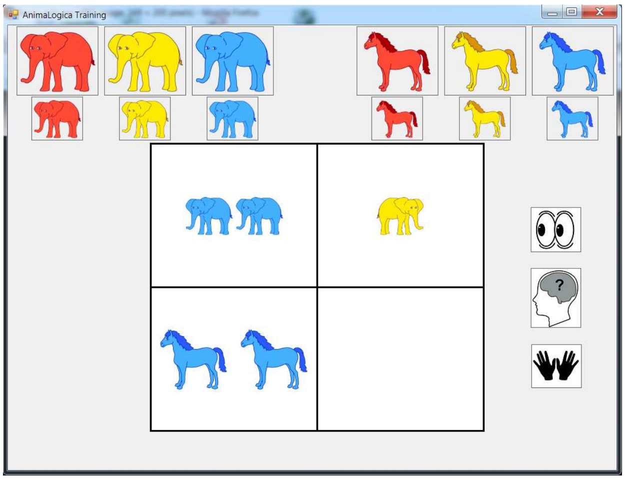


Figure 2. Depiction of visual effects emphasizes cues from prompt 1 to "Look carefully", "Think hard" and then "Try to solve the puzzle" (these are not all shown at once).

2.3.1. Feedback Interventions.

The *stepwise elaborate feedback condition* received training according to the graduated prompts method [Campione and Brown 1987; Resing and Elliott 2011] which consisted of stepwise instructions beginning with general, metacognitive prompts, such as focusing attention, followed by cognitive hints, emphasizing the transformations and solution procedure, and ending with step-by-step scaffolds to solve the problem (see Table 1). The prompts were mostly auditory in nature and accompanied by visual effects support the explanations (see Figures 2 & 3). A maximum of five prompts were administered. Once the child answered an item correctly the child was asked to explain his/her answer; no further prompts were provided and the next item was administered.

The *simple feedback condition* received auditory feedback on whether or not the outcome was correct and this was repeated until the item was solved correctly or five attempts were made to solve the item. After the fifth incorrect attempt the correct solution was shown before proceeding to the next item. If a correct solution was found before five attempts then the next item was administered.

In the *control condition* the children received the exact same items as in the other two conditions but did not receive help or feedback in solving them. Therefore, the children only practiced solving the items but were not trained in analogical reasoning.

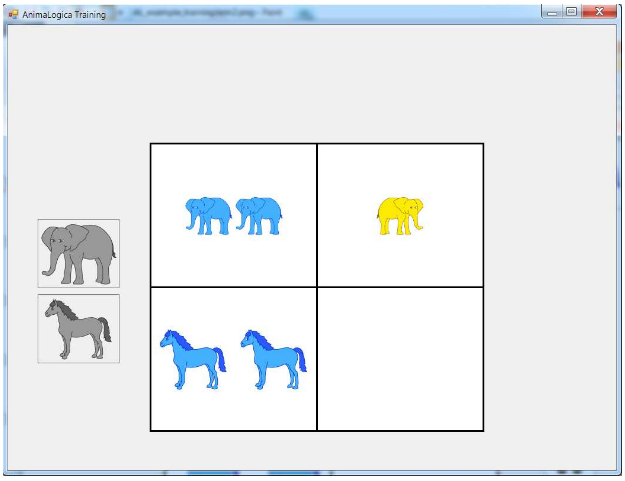


Figure 3a. Visual effects emphasizing prompt 5 where scaffolds are used to solve the puzzle: "Which animal belongs in the empty box?".

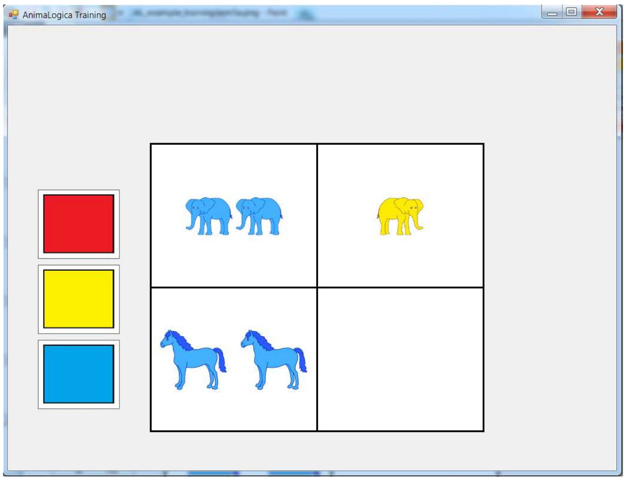


Figure 3b. Prompt 5 scaffold: "What color should it be?".

2.4 Statistical Models

Disentangling the complex changes in ability over time on an individual basis requires complex statistical models. For example, using raw gain scores (posttest minus pretest score) to measure change can lead measurement errors due to the unreliability of the gain score, the regression effect of repeated administration and that the scale units for change do not share constant meaning for test takers with different pretest scores and [de Bock 1976; Lord 1963]. These problems are potentially solved by placing ability scores for pretest and posttest on a joint interval measurement scale using logistic models such as those employed in item response theory (IRT) [Embretson and Reise 2000]. In the Rasch model, one of the most simple IRT models, the chance that an item is solved correctly depends on the difference between the latent ability of the learner and the difficulty of the presented item or problem. The Rasch-based gain score provides a good basis for the latent scaling of learning and change because the gain score has the same meaning in terms of log odds (i.e. the logarithm of probability of correct vs. incorrect) across the entire measurement scale [Embretson and Reise 2000]. Therefore, this study applied IRT models to analyze individual differences in feedback effects on learning and change [Stevenson et al. 2013a].

2.4.1 Explanatory IRT analyses

Each of the hypotheses about the children's performance and change was investigated using model comparison. First a reference model was created and then predictors were added successively to so that the fit of the new model could be compared to the previous (nested) model using a likelihood ratio (LR) test, which assesses change in goodness of fit. The models were estimated using the lme4 package for R [Bates and Maecheler 2010] as described by [De Boeck et al. 2011].

2.4.2 Null model

The initial reference model (M0) was a simple IRT model with random intercepts for both persons and items (pretest and posttest) where the probability of a correct response of person *p* on item *i* is expressed as shown in equation 1.

$$P(y_{pi} = 1 \mid \theta_p, \beta_i) = \frac{\exp(\theta_p - \beta_i)}{1 + \exp(\theta_p - \beta_i)}$$

where $\theta_p \sim N(0, \sigma_\theta^2)$ and $\beta_i \sim N(0, \sigma_\beta^2)$ (1)

2.4.3 Modelling learning and change

This study employs repeated testing. In order to account for this effect a session parameter has to be added to the null model to represent average change from pretest to posttest. However, this model assumes the effect of retesting to be equal for all children. In order to allow for individual differences in improvement from pretest to posttest a random parameter that allows for the session effect to vary over persons was added. In this model, Embretson's Multidimensional Rasch Model for Learning and Change (MRMLC, see M2 in Table 1), the chance that an item is solved correctly (P_{ip}) also depends on the difference between the examinee's latent ability (θ_p) and the item difficulty (β_i) [Embretson 1991]. Yet, the ability is built up through the testing occasions *m* up to *k* in a summation term, which indicates which abilities (θ_{pm}) must be included for person *p* on occasion *k*.

$$P(y_{ipk} = 1 \mid \theta_{pk}, \beta_i) = \frac{\exp(\sum_m^k \theta_{pm} - \beta_i)}{1 + \exp(\sum_m^k \theta_{pm} - \beta_i)}$$

where $\theta_{pm} \sim N(0, \sigma_\theta^2)$ and $\beta_i \sim N(0, \sigma_\beta^2)$ (2)

The initial ability factor, θ_{p1}, refers to the first measurement occasion (i.e. pretest) and the so-called modifiabilities (θ_{pm} with *m*>1) represents the change from one occasion to the next. In the present model examining pretest to posttest change *k*=2 and the modifiability θ_{p2} refers to performance change from pretest to posttest.

2.4.4 Modelling sources of individual differences in learning and change

The formula in equation 2 can be extended by including other item or person predictor variables and evaluating their effects on the latent scale [De Boeck and Wilson 2004]. Person predictors are denoted as Z_{pj} *(j=1,…,J)* and have regression parameters ζ_j. The item predictor (e.g. number of transformations) can be denoted as X_i (k=1) and has the regression parameter δ. These predictors are successively entered into the null model (see equation 1) as follows, with indices *i* for items, *p* for persons, *j* for the person covariate used as a predictor variable and *k* for the item covariate used a predictor variable.

$$P(y_{pi} = 1 \mid Z_{p1} \ldots Z_{pJ}, \beta_i) = \frac{\exp(\sum_{j=1}^J \varsigma_j Z_{pj} + \varepsilon_p + \delta X_{ik} + \varepsilon_i)}{1 + \exp(\sum_{j=1}^J \varsigma_j Z_{pj} + \varepsilon_p + \delta X_{ik} + \varepsilon_i)}$$

where $\varepsilon_p \sim N(0, \sigma_{\varepsilon p}^2)$ and $\varepsilon_i \sim N(0, \sigma_{\varepsilon i}^2)$ (3)

This equation represents models M3-6 in the results presented in Table 2.

Table 2.

Overview of the estimated IRT models.

Model	Nested Model	Effects			AIC	BIC	-LL	LR test[a]	
		Fixed	Random over Persons	Random over Items				*df*	*Λ*
M0			Intercept	Intercept	37575	37600	18784		
M1	M0	+ Session	"	"	35741	35775	17866	1	1835.90***
M2	M1		+Session	"	34871	34922	17429	2	874.18***
M3	M2	+ Session* Condition	"	"	34063	34132	17024	2	811.52***
M4	M3	* Strategy group	"	"	33773	33944	16866	12	314.50***
M5	M4	* WM	"	"	18014	18236	8979	8	15775***

[a] The LR-test comprises a comparison between the model and the nested model. *** p < .001

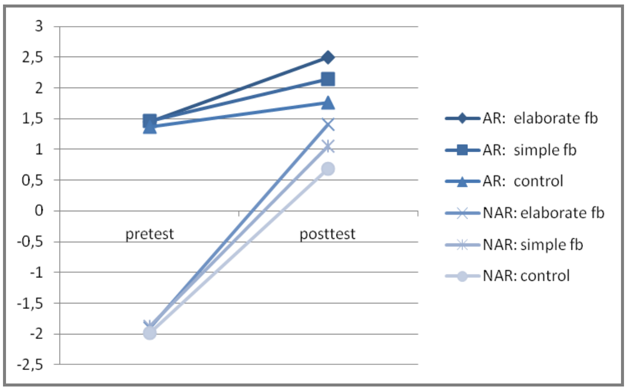


Figure 4. Plot of M5 with logit (y-axis) by Session (x-axis) for Analogical Reasoners (AR) versus Non-analogical reasoners (NAR) for each feedback condition (elaborate, repeated simple and control).

3. RESULTS

Table 2 displays the outcomes of the model building steps. As can be seen in the right-most column the addition of each new predictor in the explanatory IRT model significantly improved model fit. From M0 to M1 we could statistically infer that there was a main effect for training. The inclusion of individual regression lines for performance change from the pretest to posttest was deemed warranted given the improved model fit from M1 to M2. The significant model comparison result from M2 to M3 shows us that the different types of feedback had different "change" slopes. The difference in performance change from pretest to posttest between the two strategy-groups is shown in model M4 (see Figure 4). Finally, from M4 to M5 we could statistically infer working memory was differentially related to performance change per condition and strategy group. Analysis of the simple contrasts indicated that working memory moderated feedback effects in the analogical reasoners (AR strategy group), but was unrelated to performance change in the non-analogical reasoners (NAR strategy group) (simple feedback: B = -1.38, p <.01 and elaborated feedback: B = -1.37, p < .01, reference category = no feedback / control condition).

Significant fixed main effects were found for Session, Strategy group, verbal and visuo-spatial Working memory. Significant fixed interaction effects were found for Session x Condition, Session x Strategy group, Session x Working memory, Strategy group x Working memory and Session x Strategy group x Working memory. Random intercepts were present for persons ($SD_{ability}$ = .62, $SD_{modifiability}$ = .70, r = -.24) and items (SD = .74).

Table 3.

Estimates of fixed effects in M5.

	B	SE	*p*
Intercept	- 0.32	.42	.44
Session (reference = pretest)	2.17	.16	<.001
Simple Feedback Condition (reference = control)	0.10	.10	.32
Elaborate Feedback Condition (reference = control)	0.08	.10	.41
Strategy-group (reference = non-analogical reasoners)	3.26	.11	<.001
Verbal working memory	0.23	.09	.01
Visuo-spatial working memory	0.26	.04	<.001
Session * Simple Feedback Condition	0.28	.13	.04
Session * Elaborate Feedback Condition	0.65	.13	<.001
Session * Strategy-group	-1.65	.12	<.001
Session * Verbal Working memory	0.47	.11	<.001
Strategy-group * Verbal Working memory	0.08	.10	.43
Session * Strategy-group * Verbal Working memory	-0.61	.13	<.001

4. CONCLUSION

This paper presented our recent research in the area of statistical models of formative feedback effects in performance and change in children's analogical reasoning. The results showed that individual differences stemming from initial strategy-use and working memory efficiency were present and influenced the effect feedback. Elaborate feedback was more effective than simple feedback. Working memory was a predictor of pretest performance. Working memory also moderated feedback effects but only in children in the advanced strategy-use group. Working memory most likely forms a bottleneck in children's analogical reasoning on difficult analogy tasks [Richland et al. 2006]; however children with less advanced strategies most likely were unable to solve the more difficult analogy items which would require accurate solving steps and the accompanying greater taxation of working memory to do so. Finally, initial strategy-use interacted with feedback-type in that children using less advanced strategies at pretest benefited more from each form of feedback during training compared to the children displaying more advanced strategies at pretest. On the whole, the main conclusion is that elaborated feedback, presently implemented using graduated prompting techniques, appears to be the advisable form of feedback in advancing children's change in analogical reasoning.

Given the great potential of computer-based interactive learning environments to provide feedback tailored to an individual's instructional needs an important task is creating algorithms to optimize feedback provision and thus learning. On the one hand (meta-analyses of) randomized pretest-training-posttest control experiments that contrast the effectiveness of different types of feedback and explore sources of individual differences herein as discussed in the present paper provide essential information concerning which factors could be used to optimize feedback. However an investigation of the effects of specific elaborated feedback prompts on a trial-by-trial basis [Golden et al. 2012] and the interactions with learner characteristics or task performance (e.g., strategy-use) using item response theory models is a promising next step towards the provision of optimal feedback in interactive learning environments. Thus the next step in this research project is to expand upon the present findings concerning the effectiveness of the stepwise elaborated feedback and disentangle the immediate effects of the separate prompts during the training process. It will be interesting to see whether different types of prompts better aid more or less advanced learners with more or less efficient working memory to solve the items presented during training.

5. ACKNOWLEDGMENTS

Thank you to Marian Hickendorff for her assistance with the initial analyses of the strategy groups presented in this paper. We also thank Femke Stad and Carlijn Bergwerff for their assistance with data collection and coding.

6. REFERENCES

[1] SHUTE, V. J. 2008. Focus on Formative Feedback, *Review of Educational Research*, 78 (1), 153-189.

[2] KLUGER, A. N. AND DENISI, A.1996. The Effects of Feedback Interventions on Performance: A Historical Review , a Meta-Analysis , and a Preliminary Feedback Intervention Theory. *Psychological Bulletin*, 2(2), 254-284.

[3] NARCISS, S. AND HUTH, K. 2006. Fostering achievement and motivation with bug-related tutoring feedback in a computer-based training for written subtraction. *Learning and Instruction*, 16(4), 310-322.

[4] HATTIE, J. AND GAN, M. 2011. Instruction Based on Feedback. In *Handbook of Research on Learning*, R. E. MAYER AND P. A. ALEXANDER, eds. New York, New York, USA: Routledge, 249-271.

[5] VAN DER KLEIJ, F. M. FESKENS, R. C. W. AND EGGEN, T. J. H. M. submitted 2013. The effectiveness of methods for providing written feedback through a computer-based assessment for learning: A systematic review. 1-25.

[6] STEVENSON, C.E. 2012. *Puzzling with Potential Dynamic testing of analogical reasoning in children.*

Doctoral dissertation. Amsterdam: Leiden University, 1-191.

[7] SIEGLER, R. S. AND SVETINA, M. 2002. A microgenetic/cross-sectional study of matrix completion: comparing short-term and long-term change. *Child development*, 73(3) 793-809.

[8] TUNTELER, E., PRONK, C. M. E. AND RESING, W. C. M. 2008. Inter- and intra-individual variability in the process of change in the use of analogical strategies to solve geometric tasks in children: A microgenetic analysis. *Learning and Individual Differences*, 18(1), 44-60.

[9] HATTIE, J. AND TIMPERLEY, H. 2007. The Power of Feedback, *Review of Educational Research*, 77 (1), 81-112.

[10] CAMPIONE, J. C. AND BROWN, A. L. Linking dynamic assessment with school achievement. In *Dynamic assessment: an interactional approach to evaluating learning potential*, C. S. LIDZ, Ed. New York, New York, USA: Guilford Press, 82-109.

[11] RESING, W. C. M. AND ELLIOTT, J. G. 2011. Dynamic testing with tangible electronics: measuring children's change in strategy use with a series completion task. *The British journal of educational psychology*, 81(4), 579-605.

[12] STEVENSON, C. E., HICKENDORFF, M. RESING, W. C. M., HEISER, W. J. AND DE BOECK, P. A. L. 2013a. Intelligence Explanatory item response modeling of children's change on a dynamic test of analogical reasoning, *Intelligence*, 41(3), 157-168.

[13] STEVENSON, C. E., HEISER, W. J. and RESING, W. C. M. 2013b. Working memory as a moderator of training and transfer of analogical reasoning in children, *Contemporary Educational Psychology*, 38(3) 159-169.

[14] MULHOLLAND, T. M., PELLEGRINO, J. W. AND GLASER, R. 1980. Components of geometric analogy solution. *Cognitive psychology*, 12(2) 252-284.

[15] FREUND, P. A. AND HOLLING, H. 2011. How to get really smart: Modeling retest and training effects in ability testing using computer-generated figural matrix items, *Intelligence*, 39(4), 233-243.

[16] F. DE BOCK, 1976. Basic issues in the measurement of change. In *Advances in psychological and educational measurement*., D. N. M. DE GRUIJTER AND VAN DER L. J. T. KAMP, Eds. New York, New York, USA: Wiley.

[17] LORD, F. M. 1963. Elementary models for measuring change. In *Problems in measuring change*., C. W. HARRIS, Ed. Madison, Wisconsin, USA: University of Wisconsin Press, 21-38.

[18] EMBRETSON, S. E. AND REISE, S. 2000. *Item response theory for psychologists*. Mahwah, NJ: Erlbaum Publishers.

[19] BATES, D. AND MAECHELER, M. 2004, lme4: Linear Mixed-Effects models using S4 Classes. http://r-forge.r-project.org/projects/lme4/.

[20] DE BOECK, P. A. L., BAKKER, M., ZWITSER, R., NIVARD, M., HOFMAN, A. TUERLINCKX, F. AND PARTCHEV, I. 2011. The estimation of item reponse models with the lmer Function from the lme4 Package in R, *Journal of Statistical Software*, 39 (12), 1-27.

[21] EMBRETSON, S. E. 1991. A multidimensional latent trait model for measuring learning and change, *Psychometrika*, 56(3), 495-515.

[22] DE BOECK, P. A. L. AND WILSON, M. 2004. *Explanatory item response models: A generalized linear and nonlinear approach*. New York, New York, USA: Springer.

[23] RICHLAND, L. E. MORRISON, R. G. AND HOLYOAK, K. J. 2006. Children's development of analogical reasoning: insights from scene analogy problems. *Journal of experimental child psychology*, 94(3), 249-73.

[24] GOLDIN, I. M. KOEDINGER, K. R. AND ALEVEN, V. A. W. M. M. 2012. Learner Differences in Hint Processing, In *Proceedings of the 5th International Conference on Educational Data Mining*, Chiana, Greece, 2012.

[25] BLEICHRODT, N., DRENTH, P.J.D., ZAAL, J.N. & RESING, W.C.M. 1987. *Handleiding bij de Revisie Amsterdamse Kinder Intelligentie Test [Manual of the Revised Amsterdam Child Intelligence Test]*. Lisse: Swets & Zeitlinger.

[26] RAVEN, J. RAVEN, J.C. & COURT, J.H. 2004. Manual for Raven's Progressive Matrices and Vocabulary Scales. San Antonio, Texas: Harcourt Assessment.

[27] ALLOWAY, T.P. 2007. *Automated Working Memory Assessment (AWMA)*. London: Harcourt Assessment.

An Intelligent Tutoring System for Japanese Language Particles with User Assessment and Feedback

Zachary T. Chung
Ateneo de Manila University
Department of Information Systems and Computer Science (DISCS)
Katipunan Ave., Loyola Heights,
Quezon City, Philippines
+63-2-426-6001 local 5660
zachary.chung@obf.ateneo.edu

Hiroko Nagai, Ph.D.
Ateneo de Manila University
Japanese Studies Program (JSP)
Katipunan Ave., Loyola Heights,
Quezon City, Philippines
+63-2-426-6001 local 5248
hyabut@ateneo.edu

Ma. Mercedes T. Rodrigo, Ph.D.
Ateneo de Manila University
Department of Information Systems and Computer Science (DISCS)
Katipunan Ave., Loyola Heights,
Quezon City, Philippines
+63-2-426-6001 local 5660
mrodrigo@ateneo.edu

ABSTRACT

In recent years, an increasing number of Ateneo students have been taking an interest in the Japanese language. For Ateneo students beginning their study of the language however, Japanese particles are difficult concepts because they cannot be translated to equivalent words in English. For a beginner learner, it is inevitable to view a second language with the lens of a first language as shown by the concept of transfer in second language acquisition. As a result, learners tend to misconstrue Japanese particles by attempting to understand them with respect to non-existent equivalents in English.

In this research, we develop an intelligent tutoring system for Ateneo students taking introductory Japanese (FLC 1JSP) to aid them better understand Japanese particles. The system would assess the learner's understanding of Japanese particles by practice and depending on which particle where most mistakes are made, the system would give instructional feedback. Feedback to be implemented in the system use visual prototypes that represent the meaning of the particle. We hope to see if visual representations can also teach Japanese particles to students as an alternative to text-detailed explanations such as those commonly found in textbooks.

Categories and Subject Descriptors

K.3.1 [**Computer Uses in Education**]: Computer-assisted instruction (CAI), Distance learning

General Terms

Design, Experimentation, Human Factors, Theory

Keywords

Intelligent Tutoring Systems (ITS), Japanese particles, Case Particles, Japanese language, Visual prototypes

1. INTRODUCTION

1.1 Context of the Study

An increasing number of Ateneo students are minoring in Japanese Studies to learn more about the Japanese language and culture. Students beginning Japanese in their FLC 1JSP (Introduction to Japanese) course encounter difficulty with Japanese particles regarding proper usage and context: に（ni）、へ（e）、を（wo）、と(to)、で（de）、の（no）、は（wa）、が（ga）

1.2 Research Objectives

In this paper, we discuss the development of a web-based Intelligent Tutoring System (ITS) addressing the difficulty of Ateneo students with Japanese particles - a system that facilitates practice with feedback that clarifies particle usage and meaning. We attempt the following questions:

1. How do we create an intelligent tutoring system for Japanese to help students better understand the concept of Japanese particles?
2. Other than the topic and subject marking particles は(wa) and が(ga) respectively, which particles do students make the most mistakes with in FLC 1JSP?
3. What do these errors imply about the student's mental model of Japanese particles?

1.3 Scope and Limitations

Users of the system developed are primarily FLC 1JSP students of Ateneo de Manila University, hence system content is scoped to the said course. We aim to supplement the language knowledge of FLC 1JSP students; instruction in the system is geared towards clarifying understanding, as opposed to teaching anew.

Finally, we utilize visual feedback in the system based on prototypes by Sugimura (discussed in section 2.1) because we like to know if Japanese particles can also be taught by animations aside from explanations of their meaning. For particle and word combinations that do have not have any visual representations, we use textual feedback based on Socratic questioning as our alternative form of feedback. We hope to see if computer animations and our combination thereof can be an effective means to clarify these Japanese particles to students.

2. FRAMEWORK

2.1 Visual Prototypes for Japanese Particles

Japanese particles can be taught using images representative of their meaning. Sugimura demonstrates that each Japanese particle can be represented by a prototype image and he states that learners would have less cognitive load learning Japanese particles in this manner than rote memorization of a definition [11]. In this research, we develop visual feedback, based on five prototype images of the following particles from FLC1 JSP: **ni, e, to, no, de**.

1. The particle **ni**

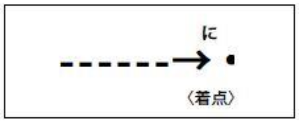


Figure 2.1: Prototypical meaning of ni [11]

Ni shows the directionality of an agent's action and its binding effect to a target [11]; **ni** can also indicate the place or time of existence of a subject [11]. These two usages are generalized into the image of a point, indicating a destination or a point in time shown above. Compared to **e**, **ni** emphasizes the destination as opposed to the process, depicted by the dotted arrow in figure 2.1.

2. The particle **de**

The particle **de** indicates *space* where an action takes place in the nominative or accusative case [11]. The prototype of this particle is shown in figure 2.2 below:

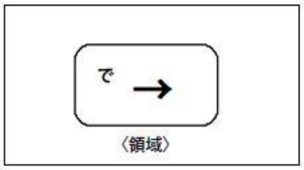


Figure 2.2: Prototypical meaning of de [11]

The arrow in figure 2.2 above represents some force acting within an enclosed space. Though **de** is likewise represented with an arrow like **ni**, **de** emphasizes an action performed within the bounds of a certain space [11].

3. The particle **e**

In essence, **e** is similar to **ni** for indicating the direction of an action. Compared to **ni**, **e** puts emphasis in the process or means of an agent to get to a destination [11; Dr. Hiroko Nagai personal communication, May 5, 2012]. The particle **e** is represented according to Sugimura in figure 2.3 below [11]:

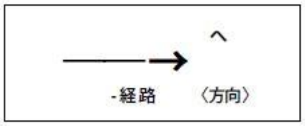


Figure 2.3: Prototypical meaning of e [11]

4. The particle **to**

According to Morita, the particle **to** has a unificative meaning associated to its usage [11], where two agents work together to perform an action. In a prototype image, Sugimura depicts the meaning of the particle **to** as follows [11] (Refer to Figure 2.4):

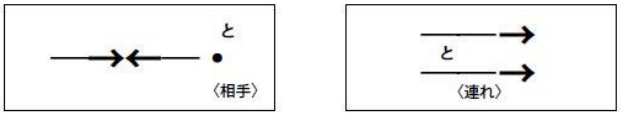


Figure 2.4: Prototypical meaning of to [11]: An action performed together in companionship.

5. The particle **no**

No denotes relations between nouns but these have various forms hence, we only consider **no** for the following usages in our research as scoped in FLC1 JSP:

1. A is the possessor of B (like the B of A or A's B) such as: watashi **no** kaban (My bag)
2. A is the location where B belongs to (B in/at A) such as: ateneo **no** gakusei (A student in Ateneo) and;
3. A created B hence B is possessed by A such as: gakusei **no** sakubun (A student's essay)

In all these three cases above, the particle **no** connects nouns together, such that the preceding noun phrase forms a phrase to modify a following noun phrase [6]. According to Oya, Japanese language adviser of the Japan Foundation, Manila, the particle **no** can be depicted in a prototypical image of a circle (noun 2) inside a larger circle (noun 1) and so on as follows for these three usages:

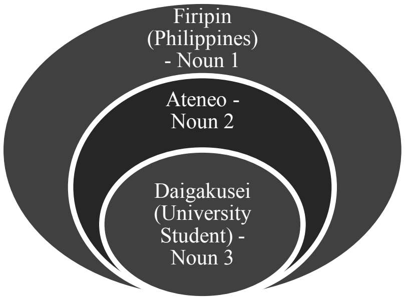


Figure 2.5: Firipin no ateneo no daigakusei: Combining nouns with no

In figure 2.5 above, the largest circle sets a scope to the circle(s) enclosed within. In this representation, Ateneo is in the Philippines and the student is affiliated with the Ateneo, thus a set of concentric circles. The enclosed nouns are connected by **no,** forming one noun, meaning "A University Student of Ateneo in the Philippines".

2.2 Visuals as Feedback in Multimedia Learning

Students learn best by seeing the value and importance of information presented so it is important to sustain interest using a feedback medium that coincides with the learning style of a student, which is "the manner in which individuals perceive and process information in learning situations" [4].

According to the Cognitive Theory of Multimedia Learning by Mayer, Multimedia instructional messages designed according to how the human mind works are more likely to lead to meaningful learning than those that are not [7]. The theory states that humans seek to make sense of multimedia presentations in relation to their collected experiences. Hence, visual feedback would be effective given that it resembles common human experience while depicting the meaning of Japanese particles. Table 3.1

summarizes the theory regarding how learners relate visuals to experience.

Table 3.1 Image-related Processes in the Cognitive Theory of Multimedia Learning: Building Connections between Pictorial Models with Prior Knowledge

Process	Description
Selecting images	Learner pays attention to relevant pictures in a multimedia message to create images in working memory.
Organizing images	Learner builds connections among selected images to create a coherent pictorial model in working memory.
Integrating	Learner builds connections between pictorial models and with prior knowledge.

As guidelines for our design of visual feedback, the following are prescribed by the theory [1, 2]:

1. **Focus on Task-Relevant Aspects of Information:** Research show that guiding learners' attention is only useful if it leads the learner to a deeper understanding of the task-relevant parts of the information presented.
2. **Limit Unnecessary Information:** Each piece of information, useful or not has to be processed by the learner so it is additive to cognitive load. According to the Apprehension Principle, information that is not required for the task or problem solving, such as seductive details or eye-catching illustrations, produce extraneous cognitive load that ties attention to less relevant concepts and therefore reduces knowledge acquisition [1].
3. **Attention-guiding Principle:** Supporting the process of selecting relevant information will be useful because it shifts the learners' attention to those parts of information that are needed to understand the key concept of presented materials. Also, since animation is fleeting by nature, often involving simultaneous display changes, it is important to guide learners in understanding the animation so that they do not miss the change. Highlights, visual cues and color coding seem to be appropriate visual instructional aids because novice learners are not able to distinguish between relevant and irrelevant features.
4. **Personalize Instruction:** Learner's attention can be activated in a more effective way if instructions are personalized rather than anonymous, for example by addressing the learner in the first person.

2.3 Error Isolation and Feedback

Mistakes are part of the learning process. According to Gass and Selinker, second language errors do not reflect faulty imitation by a learner; they are attempts to figure out a system by imposing regularity on the language being learned. In fact, mistakes are structured; there is an underlying generalization and this shows a certain level of development [3, 9].

Mistakes are akin to slips of the tongue but errors are systematic and recurring [3]. Errors mean that the learner does not recognize that it is wrong, and by consistent reproduction, he has incorporated it into his system of the target language [3]. In our system, we isolate errors by a pre-test and when an error has been committed at least twice (same particle and context), then feedback is given, targeting the faulty knowledge only as much as possible.

Feedback in our system is designed to let the learner realize his own mistake. We do this by presenting the animation of a learner's erroneous particle side-by-side with the animation of the correct particle. Alternatively, we pose questions or hints to challenge the learner to reconsider his answer instead. In this manner, we allow the learner an opportunity to explore and adjust the application of the form or rule he used to derive his wrong answer to what is correct – *restructuring* in interlanguage processes [9]. This is more effective because it does not interrupt the learner because of fear of being directly corrected [5].

3. METHODOLOGY

3.1 Development Methodology

The Intelligent Tutoring System (ITS) developed in this research is web-based for simpler deployment and testing; Adobe Flash was used to drive animations.

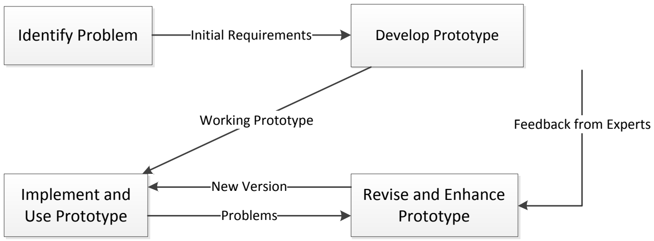


Figure 3.1: The Prototyping Methodology [8]

Based on consultations with FLC 1JSP instructors, students have difficulty mastering case particles because they confuse the different notions these particles provide in sentences. We identified particle pairs students frequently have misconceptions with such as **ni** and **de**, **to** and **no** or **ni** and **to**, etc., then we developed prototype animations that highlight their semantic differences. Then, we showed these animations to instructors for feedback and we improved them to ensure that visual feedback developed in any form teach the correct notion of Japanese particles. Consultations were performed during development mainly with Dr. Hiroko Nagai, Director of the Ateneo Japanese Studies Program, as well as with Mr. Susumu Oya, Japanese Language Adviser of the Japan Foundation, Manila, observing the processes of the prototyping methodology in software development as shown in figure 3.1 above.

3.2 Student Modeling

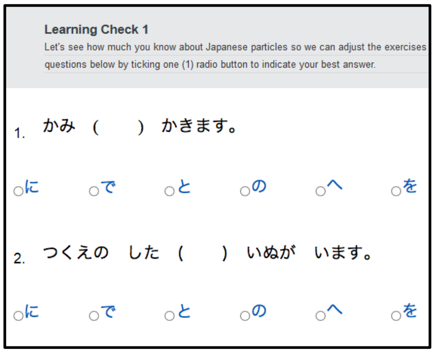
Student models provide descriptions of learning at a level of granularity that facilitates the encoding of principles and rules in a teaching system [12]. Learner models approximate student behavior by tracking misconceptions in comparison with substandard reasoning patterns. This is performed with the goal of supporting weak students' knowledge and to develop the students' strengths [13]. In our system, we used an overlay model to model the student-user of our system. The model is able to show "the difference between novice and expert reasoning, by indicating how students rate on mastery of each topic, missing knowledge and which curriculum elements need more work" [13]. Since an overlay model is a model of a proper domain subset (i.e. Japanese particles in grammar), we used this model to evaluate students and give feedback accordingly.

The disadvantage of overlay modeling is that students may have knowledge that is not part of an expert's knowledge, thus it is not represented in the student model [13]. However, we mitigate this by creating a multiple-choice based system, where possible answers are contained only within the domain knowledge we teach. Since Japanese particles also have distinct grammatical usages at the level of FLC1 JSP, creating this model is simple because the domain knowledge itself is a matter of conforming to concise grammar rules.

To create the overlay model of the student, we broke down the concept of Japanese particles from FLC1 JSP into its base knowledge components[1]. Among Japanese particles, this is the production rule learned and referenced by a learner to know how to use a Japanese particle. For example, a student can have the following knowledge component: "to indicate the existence of a living or non-living thing, the particle **ni** is used". In total, we have nine (9) knowledge components in our ITS, following a permutation of nine possible contextual usages of all the Japanese particles in our system designed for FLC1 JSP. Note that the particle **e** and the particle **ni** for indicating a place where something moves (direction) are both singly counted as one knowledge component, whereas the rest are considered as individual knowledge components. This is because FLC1 JSP does not yet teach students to differentiate the nuance of both these particles. Also, a more detailed description of how our overlay model operates is discussed below, where we also describe the general operation of the system.

3.2 General ITS System Operation Flow

Students create an account and the ITS presents a pre-test called "Learning Check 1" (See Figure 3.2). This activity shows a battery of eighteen (18) Japanese sentences using the Japanese particles taught in FLC1 JSP; the task for the student in this section is to complete the sentence by choosing the right particle to complete the statement.



Learning Check 1

Let's see how much you know about Japanese particles so we can adjust the exercises questions below by ticking one (1) radio button to indicate your best answer.

1. かみ （　　） かきます。

○に　○で　○と　○の　○へ　○を

2. つくえの　した （　　） いぬが　います。

○に　○で　○と　○の　○へ　○を

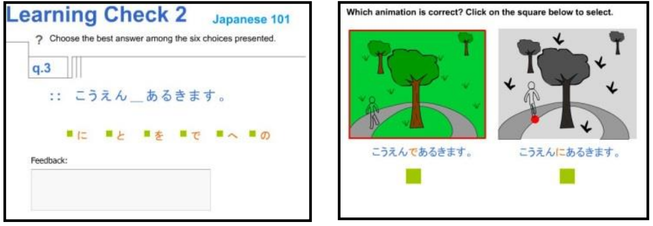
Figure 3.2: Learning Check 1 – Students complete the sentences by supplying the missing particles using the choices provided.

Learning Check 1 is used by the system to create an overlay model of the student. This is used to measure the extent of a student's knowledge of Japanese particles. The model works by assigning points per knowledge component[2] and if a student uses a particle given a context correctly, one (1) point is assigned to the corresponding knowledge component. The model works like a table, where we distribute points across rows and each row is a knowledge component. At the level of FLC1 JSP, since we have nine (9) contextual usages for the particles taught in the course and we have two questions for each usage, we have eighteen (18) questions for Learning Check 1 (See figure 3.3 below):

Pseudo-Overlay Model		
Particle	**Context**	**Pts.**
Ni	Indicate a point in time something takes place.	2
	Indicate a place where something or someone exists.	2
	Indicate target of an action by an agent (uni-directional target).	2
ni/e	Indicate a place towards which something moves.	2
De	Indicate where an event/action takes place.	2
O	Direct objects	2
No	Noun phrase modification to indicate property	2
To	Connect nouns together 'AND'	2
	Indicate target of an action by an agent (bi-directional target).	2
	Total	**18/18**

Figure 3.3 Overlay Model: Point distribution across knowledge components. Maximum attainable score is 18/18

Based on the model, the system displays content in the following section, "Learning Check 2", where actual tutoring takes place. Here, another battery of Japanese sentences is *selectively* presented about the Japanese particles the student appears to have a lack of knowledge with, had the student not met the established minimum number of points per row of the overlay model. While the student is answering, tutoring is now provided - feedback is presented on-the-fly upon mouse clicks in Adobe Flash (See Figure 3.4):



Learning Check 2　Japanese 101

? Choose the best answer among the six choices presented.

q.3

:: こうえん＿あるきます。

■に　■と　■を　■で　■へ　■の

Feedback:

Which animation is correct? Click on the square below to select.

こうえんであるきます。 ■

こうえんにあるきます。 ■

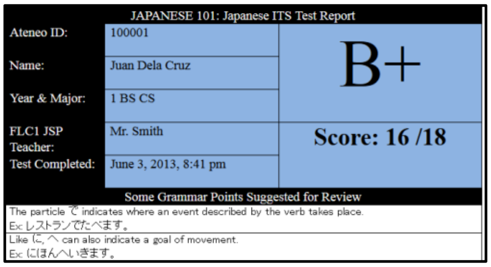
Figure 3.4: Learning Check 2 shows another sentence using 'de'; feedback as needed.

Following Learning Check 2, we present the student a post-test to measure improvements in knowledge. The post-test also serves as a follow-up learning opportunity for the student and the questions used in this section are similar to the questions in the pre-test in terms of count, particle usage and presentation but arranged in a different order. We simply changed the nouns or verbs in the

[1] A knowledge component is a process or a generalization that a learner uses alone, or in combination with other knowledge components to solve a problem [10].

[2] A knowledge component is a process or a generalization that a learner uses alone, or in combination with other knowledge components to solve a problem [7].

sentences and we also maintained two questions per context, hence also making eighteen (18) questions. This allows for comparison on an equal basis between both sections in terms of scoring. Also, to mitigate the possibility that the pre-test is more difficult than the post-test and vice-versa, we also swapped the questions we used in the pre-test with those in the post-test at random. Finally, after using the system, we show a report page to the student concluding the use of the system and how many points were earned based on the overlay model[3]. We also suggest grammar points to the student where more review is recommended based on the result of the post-test (See Figure 3.5 below).



JAPANESE 101: Japanese ITS Test Report		
Ateneo ID:	100001	B+
Name:	Juan Dela Cruz	
Year & Major:	1 BS CS	
FLC1 JSP Teacher:	Mr. Smith	Score: 16 /18
Test Completed:	June 3, 2013, 8:41 pm	

Some Grammar Points Suggested for Review

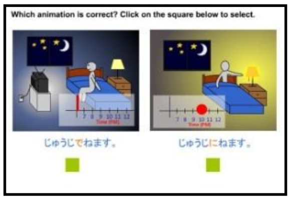
The particle で indicates where an event described by the verb takes place.
Ex: レストランでたべます。

Like に, へ can also indicate a goal of movement.
Ex: にほんへいきます。

Figure 3.5: Report Page

3.3 Feedback Design

Feedback is given by animations based on the prototype of Japanese particles (See section 2.1). For Japanese particles and their combinations thereof with certain words, forming sentences yielding an image-based representation, we show the student animations with the correct particle and the incorrect particle subtituted in the sentences side-by-side. The goal of this mode of presenting feedback is to allow the student to think for himself the correct answer before the system explicitly shows the answer with explanation. However, for cases non-illustratable, we used textual feedback based on Socratic questioning with cues. The system was designed in mind only to show explicit correction as a last resort because our goal is to restructure grammar knowledge in this tutoring system without being obstrusive to student motivation.



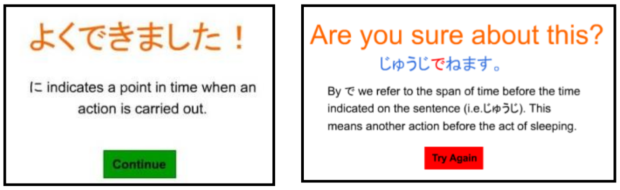
Which animation is correct? Click on the square below to select.

じゅうじでねます。 🟩

じゅうじにねます。 🟩

Figure 3.6: Animation Selection: With ‘de’ for the sentence “juuji _ nemasu (Sleep at 10pm).”, the animation of the incorrect answer (left) versus the correct answer (right) is shown.

If the student chooses the correct animation, he is praised and he is shown an explanation why his answer is correct. Otherwise, if the student still chooses the wrong animation, the system shows an explanation of the error and it allows the student to try completing the sentence again (See figure 3.6 below).



よくできました！

に indicates a point in time when an action is carried out.

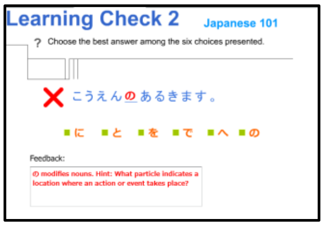
Are you sure about this?

じゅうじでねます。

By で we refer to the span of time before the time indicated on the sentence (i.e.じゅうじ). This means another action before the act of sleeping.

Figure 3.7 System Responses: Choosing the right animation leads to praise (left); choosing the wrong animation, leads to an explanation of the answer (right).

In cases when animations are not applicable, we give textual feedback in the form of clues based socratic questioning as shown in figure 3.7 below:



Learning Check 2 Japanese 101

? Choose the best answer among the six choices presented.

こうえんのあるきます。

■に ■と ■を ■で ■へ ■の

Feedback:

の modifies nouns. Hint: What particle indicates a location where an action or event takes place?

Figure 3.8: Textual feedback for syntatically impossible cases.

4. Results

4.1 Field Testing

As a system designed to target students beginning their study of Japanese in Ateneo, field testing was conducted with the aforementioned students during their FLC1 JSP classes. Students were brought to a computer lab to access the tutoring system online and a total of forty-five (45) students participated in testing across classes handled by three different instructors.

For our results in this research, we focus on presenting analysis based on the results of our pre-test versus post-test scores to see if the students improved using our ITS. Also, we evaluate the experience of the students who used our tutoring system via survey to give us an idea how they find our ITS.

4.2 Testing Methodology

Participants were divided into two (2) groups: twenty-one (21) and twenty-four (24) participants respectively. One group used the ITS such that at the onset of a mistake, corresponding feedback is already shown in Learning Check 2. Another group used the ITS such that the pair of sentences per particle and its context in Learning Check 1 must be incorrect for feedback to be given in Learning Check 2. We formed the two test groups to see how much consideration is adequate before feedback is delivered, although the latter case is ideal based on the notion of error consistency from second language acquisition. A single mistake may not necessarily translate to malformed knowledge about a concept (i.e. a mouse misclick) hence, we believe that consistency is key to isolating true faulty knowledge [3]. During testing, no student was allowed to use any references regarding Japanese particles over the internet.

[3] Each correct answer in Learning Check 1 is one (1) point. If a student commits an error, the missed points, synonymous to the number of errors made in Learning Check 1, can still be earned back provided that the student answers the corresponding follow-up questions in Learning Check 2.

Figure 4.1: Computer Laboratory Setup

4.4 Pre-test and Post-test Comparison

Table 4.1: Group 1 – One mistake, then Feedback

ID Number	Pre-test (18)	Post-test (18)	Δ
120864	8	9	1
110882	10	12	2
110966	8	4	-4
111329	6	5	-1
91388	9	13	4
122145	7	11	4
112807	10	11	1
123232	12	16	4
123653	8	10	2
123743	9	11	2
123796	9	11	2
123800	4	7	3
114162	11	12	1
94060	5	11	6
120721	10	11	1
123283	9	9	0

Table 4.2: Group 2 – Two mistakes, then Feedback

ID Number	Pre-test (18)	Post-test (18)	Δ
111662	10	11	1
114537	11	9	-2
114553	3	10	7
121314	9	14	5
121359	10	11	1
124592	10	8	-2
114512	5	9	4
110866	8	9	1
111399	11	9	-2
91957	9	8	-1
112107	3	5	2
112227	8	6	2
112017	3	5	2

In testing, we collated scores from different sections. The score in Learning Check 1 is the pre-test column. A separate post-test was carried out after Learning Check 2 to measure the change in knowledge of a student after going through the ITS.

4.5 Group 1 Analysis

For participants with a score of 13 and above in pre-testing for group 1, we did not count their results in our analysis because among all participants in this group, the highest change in score was six (6) points. This means that the highest possible improvement in points can only be measured with scores of twelve (12) and below. Students who obtained a score higher than twelve (12) can only get less than six (6) points to make it the perfect score of eighteen (18) which becomes a cap, hence there is a possibility of unequal comparison in terms of the maximum achievable improvement across students in the test group. To allow for equal and consistent comparison, these participants were excluded in the results [Dr. Joseph Beck, personal communication January 7, 2013].

All participants of group 1 found feedback in the system helpful with an average of 1.235 and 1.471 for their evaluation of the animation and textual feedback respectively on a scale of -2 to 2 (-2 as the lowest and 2 as the highest). Standard deviation values are 0.970 and 0.624 respectively for these averages. These mean that both forms of feedback used in the system are generally regarded as helpful by the participants in the group. Ease of use was evaluated by the students with an average of 1.176 and desire for a similar system for use in FLC 1JSP class was evaluated with an average of 1.294 on the same scale. Standard deviation values are 0.951 and 0.686 respectively for these averages, which point to a good consensus that the system is fairly simple to use and the students would like to have a similar system again in class. Content-wise, all the participants evaluated the system difficulty with 0.765 (from -2, easy until 2, hard) and the standard deviation is 0.437, implying that the system difficulty is manageable in terms of content. Word familiarity was evaluated with an average of 0.294 (-2 as least familiar and 2 as most familiar) with a standard deviation of 0.588. While the averages tell us that students are generally knowledgeable with the words in the system, it is neither high to indicate an excellent understanding of words nor the students are unfamiliar with the words in the system. Based on raw answers collected through the system, knowledge of words pose as a factor behind student errors because to use the correct particle, understanding the notion of words lead the decision to use the correct particle to relate them in sentences.

4.6 Group 2 Analysis

As with group 1, for students who received a score of twelve (12) and above in pretesting, we did not consider their results in our analysis to yield an equal and consistent comparison.

It appears that group 2 participants had a lower average for word familiarity at 0.000, yet the same participants found the system in terms of difficulty easier with an average of 0.615, compared to group 1 on the same scale of -2 to 2. Standard deviation values are both 0.100 and 0.650 respectively for these averages. These mean that while the participants are generally familiar with the words in the system, it also varies greatly per individual. On the other hand, system difficulty is moderate for the participants of this group. Notably, lower averages were attained with 0.667 and 1.083 regarding feedback helpfulness in animation and text respectively. The standard deviations for these values are 0.778 and 0.669 respectively. Ease of use and desire for use of the system in FLC1 JSP gained lower averages at 0.846 and 1.077 with standard deviations values of 0.689 and 0.641 respectively. For these lower scores, it is possible that because participants received feedback less in this group, they found the system less helpful hence more difficult.

5. Conclusion

Table 5.1: Average Delta in Scores (Pre-test vs. Post-test)

1 Mistake (Group 1)	1.75pts.
2 Mistakes (Group 2)	1.38pts.

Findings show that the ITS is effective for both test groups as shown by the positive increase in average delta scores for both test groups. However, more aggressive feedbacking seem to lead to a better perception of the ITS and higher improvement in scores among participants are evident in group 1 than in group 2. In computer-based teaching, it appears that immediate feedback is better whenever an error is committed at the onset, contrary to what we posited based on concepts in second language acquisition, where it is best to wait for consistent error production first before feedback. In classroom-based teaching, direct correction is not advised, however in computer-based teaching where correction is already indirect by nature through a screen and not by person, immediate correction is more effective and best at the onset of an error.

As initial work in the field, much improvement can still be done to further this ongoing research. In consultation with Dr. Joseph Beck, a visiting professor from Worcester Polytechnic Institute, he suggests to add follow-up questions with our animations, confirming if the user did understand what is taught by the system right after any feedback. Also, from theory to our direct application of image-based teaching of Japanese particles by Sugimura, more investigation regarding effective visual feedback design could be carried out because how we translated the theory into animation based on theoretical meaning may not deliver the intended idea of what we mean to show the student. By doing so, it is possible to uncover the elements in animated feedback students find particularly helpful regarding these particles. From this endeavor, we know that an effective intelligent tutoring system centered on animations for Japanese particles works when it guides the self-discovery learning of students. Success is notable when the students themselves can reproduce the correct answer on their own on a similar question immediately after feedback.

Finally, to have a more in depth understanding of the causality of learner errors and to further confirm our analysis regarding trends among these Japanese particles, we plan to conduct follow-up interviews with select participants to factor in how a user understands certain aspects of the system in relation to a participant's understanding of Japanese.

6. ACKNOWLEDGMENTS

Our thanks to the Ateneo Laboratory for the Learning Sciences; the Department of Science and Technology Philippine Council for Industry, Energy, and Emerging Technology Research and Development (PCIEERD) for the grant entitled, "Development of Affect-Sensitive Interfaces"; the Engineering Research and Development for Technology (ERDT) program for the grant entitled, "Development of an Educational Data Mining Workbench."; Joseph Beck, Ph.D. from Worcester Polytechnic Institute and Susumu Oya from the Japan Foundation.

7. REFERENCES

[1] BETRANCOURT, M. 2005. The animation and interactivity principles in multimedia learning. In The Cambridge Handbook of Multimedia Learning, R.E. Mayer, Ed. Cambridge University Press, NY, 287-295.

[2] BRUNKEN, R. AND PLASS, J. AND LEUTNER, D. 2002. How instruction guides attention in multimedia learning. In Proceedings of the 5th International Workshop of SIG 6 Instructional Design of the European Association for Research on Learning and Instruction (EARLI), Eurfurt, June 2002, H. NEIGEMANN, D. LEUTNER AND R. BRUNKEN, Eds. Waxmann Munster, Munchen, Berlin, 122-123.

[3] GASS, S.M. AND SELINKER, L. 2001. Second language acquisition: An introductory course, 2nd ed. Lawrence Erlbaum Associates, Mahwah, NJ.

[4] GILAKJANI, A.P. AND AHMADI, S.M. 2011. The Effect of Visual, Auditory and Kinesthetic Learning Styles on Language Teaching. In International Conference on Social Science and Humanity, Singapore, February, 2011. IACSIT Press, Singapore. 469-472.

[5] KODAMA, Y. AND KIDA, M. 2010. Teaching grammar. In Nihongo Kyoujuho Series, Japan Foundation. Sanbi Printing, Bunkyo-ku, Tokyo. 25-32.

[6] MAKINO, S AND TSUTSUI, M. 1989. A dictionary of basic Japanese grammar. The Japan Times, Tokyo, Japan.

[7] MAYER, R.E. 2005. Cognitive theory of multimedia learning. In The Cambridge Handbook of Multimedia Learning, R.E. Mayer, Ed. Cambridge University Press, NY, 32-43

[8] NAUMANN, J.D. AND JENKINS, A.M. n.d. Prototyping: The new paradigm for systems development. MIS Quarterly, 6 (3). 29-44.)

[9] ORTEGA, L. 2009. Understanding second language acquisition. In Understanding Language Series, B. Comrie and G. Corbett, Eds. Hodder Education, London, United Kingdom. 116-118.

[10] PITTSBURGH SCIENCE OF LEARNING CENTER: LEARNLAB. 2011. http://www.learnlab.org/research/wiki/index.php/Knowledge_component

[11] SUGIMURA, T. 2002. Teaching each Japanese particles through images (イメージで教える日本語の格助詞, trans.). Language Culture Research Series. Nagoya University International Language and Culture Research Center, Japan. 39-55.

[12] WOOLF, B.P. 1992. Towards a computational model of tutoring. Educational Technology Research and Development, 40. (4) 49-64.

[13] WOOLF, B.P. 2009. Building intelligent interactive tutors: Student centered strategies for revolutionizing e-learning. Morgan Kaufmann Publishers, Burlington, MA

Towards Formative Feedback on Student Arguments

Nancy L. Green
University of North Carolina
Greensboro
Greensboro, NC 27402 USA
1-336-256-1133
nlgreen@uncg.edu

ABSTRACT

This paper presents our ideas on generating formative feedback in the Genetics Argumentation Inquiry Learning (GAIL) system. GAIL will provide undergraduate biology students with tools for constructing Toulmin-style arguments on questions in genetics. Feedback will be based in part on the output of GAIL's argument analyzer, which will compare learner arguments to automatically constructed expert arguments. In addition to identifying problems in the learner's arguments, the analyzer will recognize the argumentation scheme used to construct acceptable arguments. From that, GAIL can instantiate critical questions, a unique form of feedback in intelligent learning environments.

Keywords

Educational Argumentation Systems, Undergraduate Genetics Education.

1. INTRODUCTION

We are developing the Genetics Argumentation Inquiry Learning (GAIL) system for improving undergraduate biology students' argumentation skills in the domain of genetics. As in many educational argumentation systems, GAIL will provide the learner with tools for representing arguments in diagrams due to the cognitive benefit of diagrams [1-3]. In addition, educational systems can exploit the learner's argument diagram as a source of information for providing educational feedback. A prototype graphical user interface (GUI) for GAIL is shown in Figure 1. The top left-hand side of the screen presents a problem, e.g., to make an argument for the claim that J.B., an imaginary patient, has the genetic condition called cystic fibrosis. Below that are possible hypotheses, data about the patient and his biological family members, and biomedical principles that may be relevant to the current problem. The learner can drag these elements into the argument diagramming workspace in the center of the screen to construct an argument in a Toulmin-influenced [4] box-and-arrow notation; a vertical arrow from the *data* points upward to the *claim/conclusion* and the *warrant* is attached at a right-angle to the arrow.

In this paper we describe our planned approach to providing formative feedback based upon automatic analysis of learners' argument diagrams. Expert models for argument analysis will be automatically constructed by GAIL using an argument generator module similar to the argument generator developed for the GenIE Assistant [5]. The expert model will contain all acceptable arguments that can be generated automatically for a given claim from an underlying knowledge base (KB) representing the problem domain. GAIL's argument analyzer will compare the user's argument to the generated expert arguments to identify acceptable learner arguments and weaknesses in the learner's argument. Weaknesses in student arguments are identified using non-domain-specific, non-content-specific rules that recognize common error types, e.g., those observed in a pilot study reported in section 3. In addition, if an argument is acceptable, the analyzer will recognize and output the argumentation scheme underlying the student's argument and its associated critical questions. The output of GAIL's argument analyzer will be utilized by GAIL's feedback generator to provide formative feedback.

In some previous educational argumentation systems, the student's argument diagram is compared to a manually-constructed expert model to provide problem-specific support. However, expert models are expensive to construct and may not cover all possible solutions or errors [6]. In GAIL's approach the expert model is constructed automatically. Other systems use simulation of reasoning to evaluate formal validity but do not provide problem-specific support [6]. GAIL's approach is similar in that it reasons like an expert to generate an argument. Unlike those systems, however, GAIL's approach will provide problem-specific support.

This paper presents how the expert model is generated (section 2), a pilot study of GAIL's GUI prototype that motivated the classification of weaknesses in learners' arguments (section 3), implementation of a prototype argument analyzer (section 4), some issues to be addressed in the planned feedback generator (section 5), and conclusions (section 6).

2. EXPERT MODEL

Generation of expert arguments in GAIL will be done following the approach to argument generation used in the GenIE Assistant, a proof-of-concept system for generating first-drafts of genetic counseling patient letters [5]. Written by genetic counselors to their clients, this type of letter contains biomedical arguments to justify diagnostic testing, the diagnosis of genetic conditions, and the probable genotypes of family members. GenIE's internal components include

- *domain models*, causal models of genetic conditions used by genetic counselors in communication with their clients [7],
- an *argumentation engine* that uses computational definitions of *argumentation schemes* [8] to guide search in the domain model for data and warrant needed to support a particular claim, and
- a *letter drafter* that organizes and expresses the arguments as English text using natural language generation techniques.

GAIL's expert arguments will be produced using a similar approach to the GenIE Assistant's domain models and argumentation engine. However, the natural language generation module, the letter drafter, will not be needed to generate expert arguments.

The domain models in the GenIE Assistant are represented computationally as qualitative probabilistic networks (QPN) [9]. A QPN consists in part of a directed acyclic graph whose nodes are random variables. In addition, a QPN specifies qualitative constraints on variables in terms of influence (S^+, S^-), additive synergy (Y^+, Y^-), and product synergy (X^0, X^-) relations. For (Boolean) random variables A, B and C, $S^+(A,B)$ [or $S^-(A,B)$] can be paraphrased as *If A is true then it is more [less] likely that B is true*; $Y^+(\{A,C\},B)$ *[or* $Y^-(\{A,C\},B)$ as *If A and C are true then A enables [prevents] C from leading to B being true;* $X^0(\{A,C\},B)$*[or* $X^-(\{A,C\},B)]$ as *if both [either] A and C are true then it is likely that B is true*.

To illustrate S^+, if a patient has two mutated BRCA1 alleles then it is more likely she will develop breast cancer; $Y+$, someone who has inherited a genetic mutation for familial hypercholesterolemia is at a higher risk of heart disease if she is obese; X^-, breast cancer can be caused by mutation of BRCA1 or some other gene; and X^0, together the mother and the father can pass an autosomal recessive mutation to their offspring. A QPN representing knowledge about a genetic condition can be reused for different patient cases. Representative domain models for testing the GenIE Assistant were built quickly using information from genetics reference books. The size of a QPN to be used in GAIL would be of the same scale as those used to generate letters in the GenIE Assistant (less than 50 nodes). For more information on domain modeling see [5].

Computational definitions of argumentation schemes are used by the GenIE Assistant's argumentation engine to construct a genetic counselor's arguments for the diagnosis and genotypes of family members [5]. The argumentation schemes are formalized in a structure including *claim*, *data*, and *warrant*. Since the argumentation engine and schemes do not encode domain-specific or patient case-specific content, they can be used to generate arguments in any domain whose domain knowledge can be represented in a similar format. The propositions used as claim or data describe states of variables in a QPN. The warrant expresses formal constraints on the nodes of the QPN in terms of influence and synergy relations mentioned above. The distinction between the two types of premises reflects their difference in function and source of information. Claims and data are facts or hypotheses about a particular case, whereas warrants describe (biomedical or other) generalizations.

In addition to those components, argumentation schemes in the GenIE Assistant include a field called the *applicability constraint*, a constraint that must be true to generate an argument from that scheme. Note that conclusions of the argumentation schemes are not necessarily deductively valid, and the *applicability constraint* is a type of critical question [8]. As discussed in section 5, the *critical questions* of GAIL's argumentation schemes provide a systematic means of challenging the conclusion of an argument.

To illustrate, consider an abductive reasoning scheme used in the GenIE Assistant:

Claim: $A \geq a$
Data: $B \geq b$
Warrant: $S^*(<A,a>, <B,b>)$
App. constraint: $\neg$ *exists* $C\ X^-(\{C,A\},<B,b>): C \geq c$

In the above, uppercase-initial terms -- *A, B, C* -- are random variables in the QPN, S^* is a chain of one or more positive influence relations S^+. Lowercase-initial terms – *a, b, c* – are values of the random variables, and in this scheme are threshold values. To paraphrase this scheme, (warrant) there is a (chain of) possible positive causal influence(s) from *A* to *B*; (data) *B* is at least *b*; therefore (claim) *A* is at least *a*; (applicability constraint) provided that there is no *C* such that *C* and *A* are mutually exclusive positive influences on *B* and *C* is at least *c*. For example, (warrant) having a genotype with two mutated alleles of CFTR can lead to (abnormal CFTR protein which can lead to abnormal pancreas enzyme level which can lead to) growth failure; (data) this patient has growth failure; therefore (claim) this patient has cystic fibrosis; (applicability constraint) as long as there is no other condition believed to explain growth failure.

An argument for a given claim is automatically constructed by searching the domain model and data about the patient's case for information fitting GenIE's argumentation schemes instantiated with the claim. In addition to the above abductive argumentation scheme, other schemes support abductive reasoning about alternative causes or jointly necessary causes, reasoning from cause to effect, reasoning from negative evidence, and reasoning by elimination of alternatives. The argumentation schemes reflect those used in a corpus of genetic counselor-authored letters. Note that the GenIE Assistant's argumentation engine can construct complex arguments involving multiple pieces of evidence and chains of arguments. The same approach will be used in GAIL to generate expert arguments for a given claim. In a performance evaluation of the GenIE Assistant, two letters, each containing multiple arguments, were generated in 22 seconds on a desktop computer [5]. Note that the time should be less than that in GAIL, since the arguments will not be realized in English. Also, they can be generated off-line if necessary.

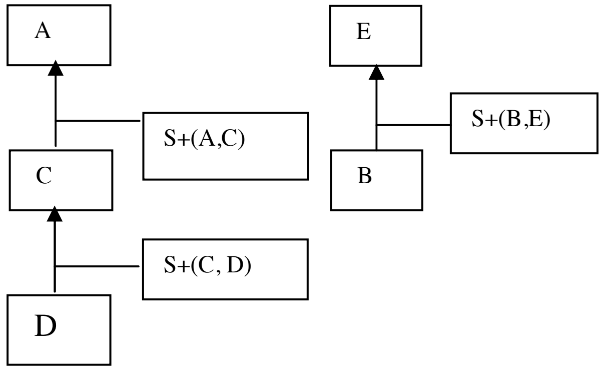


Fig. 2. Example of simple argument structures.

Some example arguments that can be generated are illustrated in Figures 2 and 3 in the box and arrow style of notation used in the GAIL interface. (To save space, the diagrams contain variables rather than the text that would be used in the GUI.) The diagram on the left of Figure 2 is a chain of two abductive arguments. The claim (A) that patient P has cystic fibrosis (two mutated CFTR alleles) is supported by the hypothesis (C) that P has abnormal CFTR protein and is warranted by the positive influence relation between CFTR alleles and CFTR protein. Hypothesis C is supported by the data (D) that P has frequent respiratory infections and the positive influence relation between CFTR protein and respiratory infections. The diagram on the right of

Figure 2 is a causal/predictive argument for the claim (E) that individual M (the patient's mother) is a carrier of a CFTR mutation. E is supported by the family history data that M has a certain ethnicity and is warranted by the higher probability of being a carrier if an individual has that ethnic background.

Figure 3 shows part of an argument for the claim (A=1) that P's mother has exactly one mutated CFTR allele. The left-hand subargument is for the hypothesis that she has one or two mutated CFTR alleles. That subargument is supported by the hypothesis (D=2) that P has cystic fibrosis (two mutated CFTR alleles), and is warranted by the synergy relation, X^0(<A=1,B=1>, D=2), i.e., that a child who has two mutated alleles inherited one from the mother and one from the father. Note that the claim D=2 would be supported by another subargument (not shown in Figure 3). The right-hand subargument is for the hypothesis that the mother does not have two mutated CFTR alleles. This is supported by the data (¬C) that she does not have cystic fibrosis symptoms, and warranted by the positive influence relation between CFTR alleles and symptoms of cystic fibrosis.

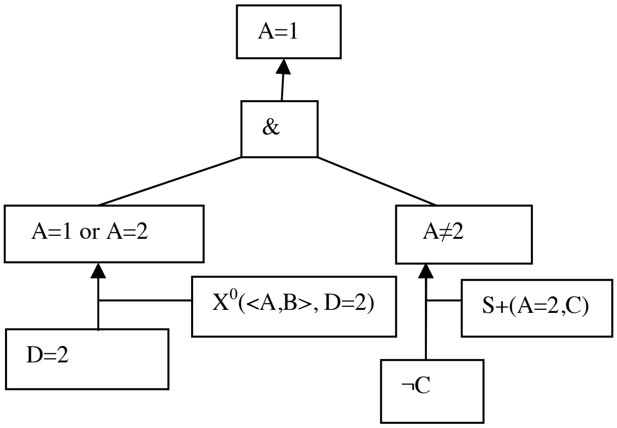


Fig. 3. Example of part of more complex argument.

3. PILOT STUDY

A formative evaluation of GAIL's prototype user interface was done in fall 2011 through spring 2012 with a total of 10 paid undergraduate volunteers, the first seven of which were recruited from biology classes and the last three computer science students. Each participant was first asked to read a seven-page patient education document, which we had found on the internet and printed for this study, on the inheritance and diagnosis of cystic fibrosis. After a participant read the document, it was put away and the research assistant narrated a silent video tutorial describing the components of an acceptable argument, and showing the features of the GAIL GUI and the process of constructing several different arguments using GAIL. Afterwards, the research assistant pointed out a chat box in the GAIL GUI for communicating with the assistant if necessary. The assistant then left the room, but could view the participant's computer screen on another computer monitor.

Listed in the upper left-hand corner of the GAIL GUI, the problems for which the first seven participants were asked to construct arguments are as follows.

Problem 1: Give two arguments for the diagnosis that J.B. has cystic fibrosis.

Problem 2: Give one argument for the diagnosis that J.B.'s brother has cystic fibrosis.

Problem 3: Give one argument against the diagnosis that J.B.'s brother has cystic fibrosis.

Problem 4: Give one argument for hypothesis that J.B.'s mother and father are both "carriers" of the CFTR gene mutation that causes cystic fibrosis

Note that the hypotheses, observations, generalizations (warrants), and problems shown on GAIL were written by the author of this paper based on information from a college genetics textbook. (J.B. refers to a fictitious patient.)

None of the first seven students created acceptable arguments. At that point in the study, it was decided to modify the materials and procedure. First, the problems were reduced in number (eliminating Problem 2, requiring an argument with conjunction). Second, when the participant submitted a response, the research assistant reviewed it using a checklist of error types created by the author after reviewing the arguments created by the first group of participants. If the participant's response contained any of those types of errors then the research assistant gave the participant feedback (as discussed below) through the chat box and asked the student to revise his argument. After three tries, the student was told to proceed to the next problem in the set. Third, to expedite the revised study, the remaining three students were recruited from computer science.

The distribution of error types is shown in Table 1. A Type 1 error was an argument whose claim did not match the claim for which the student was asked to give an argument. Type 2 was an argument where the data was not evidence for the claim. Type 3 was an argument where the warrant did not relate the data to the claim. Type 4 was an argument where the opposite type of link was required. Type 5 was a chained argument in which a subargument was missing or incorrect. For example, consider the chained argument on the left of Figure 2. If the learner failed to give a subargument in support of C, or if the learner skipped the intermediate conclusion C and showed D as directly supporting A, the error would be classified as Type 5. Type 6 errors involved incorrect use of conjunctions. Type 7 was omission of the warrant.

Table 1. Average number of errors per error type per person in each group

Error Type	Group 1	Group 2
1:Incorrect claim	1.9	0.8
2:Incorrect data	2.6	0.3
3:Incorrect warrant	2	1
4:Incorrect pro/con	0.9	0.3
5:Incorrect/missing chained claim	1.4	0
6:Incorrect/missing conjunction	0.9	NA
7: Missing warrant	0.1	0.4

In Table 1, Group 1 comprises the first seven students, who were given no feedback. Group 2 comprises the last three students, who were given feedback and three tries on each problem. The number of errors on each try for each student in Group 2 was totaled and the average was computed by dividing by nine (i.e., three students with three tries each). From the first group, it can be seen that the most frequent errors (in descending frequency) were incorrect data, incorrect warrant, and incorrect claim. Although the quantity of errors in the first and second groups cannot be compared, it should be noted that the top three error types in Group 1 remained the top three in Group 2.

Group 2 received feedback from the research assistant based on the following guidelines:

1. Does the hypothesis match the problem? If not, tell the student that the hypothesis must match the problem.
2. Is everything OK except that the student has used Pro instead of Con or vice versa? If so, explain the difference.
3. Is the data relevant to the hypothesis (could you make a good argument using that data)? If not, suggest he/she try to use some other data.
4. Is the data relevant but the generalization (warrant) does not link the data to the hypothesis? If yes, suggest he/she try a generalization that links the two.
5. Is the generalization (warrant) relevant (could you make a good argument with it) but the data does not fit the warrant? If yes, suggest that he/she try different data that fits the warrant.
6. Did the student include some data in a conjunction that is unnecessary? If so, suggest that he/she remove the conjuncts that do not fit the warrant.
7. Did the student appear to skip a step in a chained argument that has a sub-argument for the data of the top argument? If yes, help the student break it into the main argument and the sub-argument.

Table 2 shows the types of errors made by the three students in Group 2 after receiving feedback on their first and second answers on each problem. Problem 1 was solved correctly by two students on the first try, and by the third student on the second try. Problems 2 and 3 were solved correctly by only one student (on the third try). Problem 3 was solved correctly by two students on the second try. These results suggest that on the more difficult problems (Problems 2 and 3), the feedback may have helped to reduce the number of errors.

Table 2. Types of errors in group 2 (after feedback).

Student	Try	Problem 1	Problem 2	Problem 3
1	1st		1, 3 ,4	2, 3
	2nd		1, 3	7
	3rd		3, 4	2, 7
2	1st	1	1, 3	1, 7
	2nd		1, 3	
	3rd		1	
3	1st		3, 4	2, 3, 7
	2nd		3	
	3rd			

At the end of the session, students were asked to complete a user experience survey. The survey results, shown in Table 3, indicate that the students had a favorable response to using the software despite making errors.

Table 3. Average scores on user experience survey (N=10). Possible responses: 3(True), 2(Somewhat true), 1(False).

Question	Score
My background … helped me answer the problems in this study.	2.3
I found the subject of genetic conditions and inheritance interesting.	3
I found the tools for diagramming arguments easy to use.	2.8
I found the tutorial on how to use the argument diagramming tools helpful.	3
I prefer using the argument diagramming tools to writing arguments.	2.7
I would like to use a program like this in my courses on genetics	2.9

4. ARGUMENT ANALYZER

The expert model will contain all acceptable arguments that can be automatically generated for a given claim from an underlying knowledge base (KB) representing the problem domain. The generated arguments are simple or complex argument structures containing KB elements. Text elements provided to the learner through GAIL's GUI are linked internally to KB elements. The inputs to GAIL's argument analyzer will be the learner's argument and the expert model, both in the same format. Implemented in Prolog, the prototype argument analyzer determines if a student's argument diagram represents an acceptable argument and if not acceptable, identifies its weaknesses.

The algorithm to determine acceptability merely checks whether the user's argument matches one of the acceptable arguments. If the user's argument does not match an acceptable argument, its weaknesses are identified using pattern-matching rules motivated mainly by the types of errors seen in the study described in the previous section. The rules are non-domain-specific and non-problem-specific. For example, if the user's data and claim match the expert's, but the warrant does not, the analyzer identifies the problem as an unacceptable warrant (Type 3). The prototype argument analyzer implementation outputs an error message for each error detected. However, in the future implementation of GAIL, the argument analyzer's output would be used by the Feedback Generator, which will be responsible for selecting which error(s) to highlight and providing appropriate feedback.

If the learner's argument is acceptable, i.e., it matches an expert argument, then knowledge of the argumentation scheme used to generate the expert argument provides an additional resource for generation of feedback as described in the next section.

5. FEEDBACK GENERATOR

The feedback generator has not been implemented yet. Currently, we are gathering information to guide its design. As discussed in the previous section, the feedback generator will have access to the output of the argument analyzer. If the learner's argument contains errors such as those types listed in Table 1, some design questions are: which of the errors to address (and in what order), when to provide feedback, what feedback content to provide, and in what syntactic form. Before designing a feedback generator that answers these questions, we are running a think-aloud study to get a better understanding of why students make these errors. For example, a type 4 error might be due to a misunderstanding of the argument representation used in GAIL's GUI. If that is indeed the case, then it would seem that addressing such an error should be given higher priority by the feedback generator. On the other hand, we hypothesize that a type 1, 2 or 3 error may be due to a deeper problem, either in the learner's understanding of what constitutes an acceptable argument, or in understanding the genetics information provided by GAIL as possible building blocks for the learner's argument diagram.

A key point to note is that our approach supports content-based feedback. Many of the types of errors listed in Table 1 are content-based errors that can be detected by the argument analyzer based on the expert model. In addition to using it to identify content-based errors, GAIL will be able to use the expert model to provide content-based feedback. This is illustrated in the following imaginary scenario. Figure 4 depicts abstractly a student argument diagram in which the data, B, is not related by the warrant, S+(A,C), to the conclusion A. Our approach supports providing feedback to the effect that this argument is not acceptable because the warrant does not relate the data to the conclusion; and supports giving the advice to look for other data that is consistent with the given warrant or to look for another warrant that links the given data to the conclusion. Suppose that the expert model contains an argument similar to that in Fig. 4, but using C as data. If the student is unable to make use of the more general advice to replace the data or warrant in the diagram, a hint could then be generated asking whether C is in the observations or hypotheses on the GUI screen.

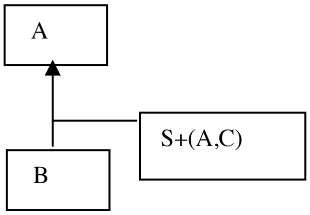


Fig. 4. Abstract example of unacceptable argument.

Figure 5 shows that with the help of this feedback, the imaginary student has replaced the data in the argument diagram with C. However, suppose that C was listed on the GUI screen as a hypothesis rather than an observation. In that case, a sub-argument for C would be required. The argument analyzer could recognize that the sub-argument for C in the expert model is missing in the student's diagram. Then the feedback generator could inform the student that C must be supported by a sub-argument since it is only a hypothesis.

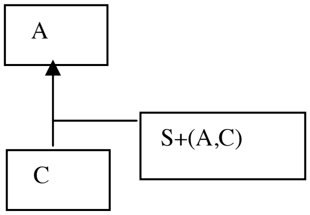


Fig. 5. Abstract example of partly fixed, unacceptable argument.

Figure 6 shows that with the help of this feedback the student adds a sub-argument for C to the diagram, matching an acceptable expert argument.

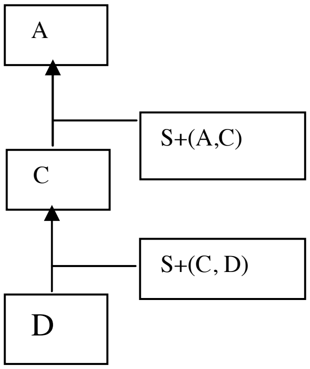


Fig. 6. Abstract example of acceptable argument.

In this domain, however, the conclusions of acceptable arguments are not necessarily deductively valid. As discussed in Section 2, each abstract argumentation scheme is associated with certain critical questions, which provide a way of challenging an argument constructed from that scheme. Critical questions support a different type of feedback, which could inspire a learner to consider multiple arguments pro and con the same claim. To illustrate, one of the critical questions of the abductive argumentation scheme is whether there is another plausible explanation of a certain observation. Having recognized the learner's argument as an instance of this scheme, the feedback generator could instantiate this critical question. Suppose that the learner has constructed an acceptable abductive argument for a diagnosis of cystic fibrosis; instantiating this critical question could support generating feedback such as *Can you make an argument for an alternative diagnosis that explains the patient's frequent respiratory infections?* or, *What if he has some other condition that could explain those symptoms?*

Some other critical questions of GAIL's abductive argumentation schemes, where B is an observation and A is a putative cause of B, include (Green 2010):

- **(Missing Enabler)** is there a C such that C is required for A to cause B, and C is absent? (Example: *Has exposure to bacteria occurred, which is required for thickened mucous to lead to frequent respiratory infections?*)
- **(Mitigation)** is there a C whose presence may mitigate the effect of A on B? (Example: *Is the patient taking antibiotics, which will prevent respiratory infections?)*
- **(Inapplicable Warrant)** Despite the similarity of individual I to the population described by the warrant, is there is a difference that could make it inapplicable to I? (Example: *Although the mother is from a geographic region with a high rate of cystic fibrosis, is her ethnic background different from most of the population there?*)
- **(False Positive)** Is p(¬A | B) too high? (Example: *Is the false positive rate for the laboratory test used to diagnose this condition high?*)
- **(Low Certainty of Data)** Is p(B) too low? (Example: *Are we confident that there is accurate information about the health of the biological mother who gave the patient up for adoption when he was an infant?*)

Again note that feedback can be given without requiring problem-specific knowledge to be embedded in the feedback generator. Also note that semantic, not syntactic, forms of critical questions are associated with argumentation schemes. Thus, using natural language generation from semantic forms to generate syntactic variations, one could study the varying effectiveness of different ways of asking the same critical question.

6. CONCLUSIONS

This paper presents our ideas on generating formative feedback in the Genetics Argumentation Inquiry Learning (GAIL) system. GAIL will provide learners with tools for constructing Toulmin-style arguments in diagrams using blocks of text provided by the system. The text is linked internally to KB elements. An argument generator like one previously developed for another application will use the KB and abstract argumentation schemes to automatically generate expert arguments. GAIL's argument analyzer will determine if a learner's argument is acceptable by comparing it to the expert arguments. A prototype argument analyzer has been implemented using non-domain-specific, non-content-specific rules that recognize common error types. The error types are based on those observed in a pilot study. GAIL's formative feedback generator will use the argument analyzer's output. In addition to identifying problems in the learner's argument, if the argument is acceptable the analyzer will inform the feedback generator of critical questions of the argumentation scheme underlying the student's argument. The critical questions can be used to generate feedback stimulating the learner's critical thinking.

7. ACKNOWLEDGMENTS

Graduate students B. Wyatt and C. Martensen implemented the prototype of GAIL's GUI in summer 2011, and Martensen ran the user study in fall 2011 through spring 2012; both received support from a UNCG Faculty Research Grant. We would like to thank the reviewers as well for asking many interesting questions that we have tried to address in the camera-ready version of this paper or would like to address in future work.

8. REFERENCES

[1] Kirschner et al. 2003. Kirschner, P.A., Buckingham Shum, S.J., and Carr, C.S. (Eds.) 2003. *Visualizing Argumentation*. London: Springer.

[2] Scheuer et al. 2010. Scheuer, O., Loll, F., Pinkwart, N., and McLaren, B.M. 2010. Computer-Supported Argumentation: A Review of the State of the Art. *Computer-Supported Collaborative Learning* 5(1): 43-102.

[3] Pinkwart and McLaren 2012. Pinkwart, N., McLaren, B.M. (Eds.) 2012. *Educational Technologies for Teaching Argumentation Skills*. Sharjah: Bentham Science Publishers

[4] Toulmin, S. 1998. Toulmin, S.E. 1998. *The uses of argument*, Cambridge: Cambridge University Press.

[5] Green, N., R. Dwight, K. Navoraphan, and B. Stadler. 2011. Natural language generation of transparent arguments for lay audiences. *Argument and Computation*, 2(1): 23-50.

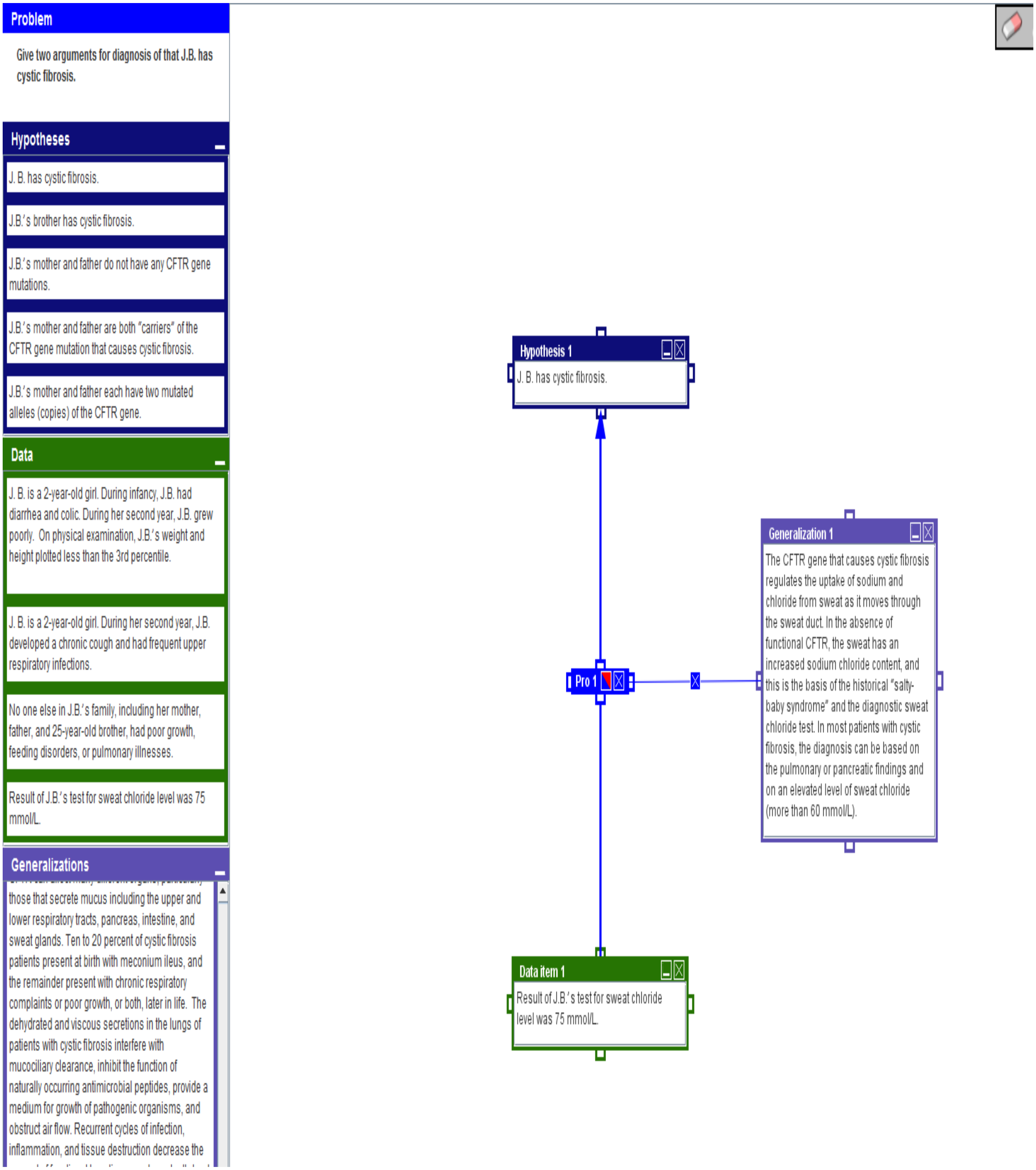
[6] Scheuer et al. 2012. Scheuer, O., McLaren, B.M., Loll, F., Pinkwart, N. 2012. Automated Analysis and Feedback Techniques to Support and Teach Argumentation: A Survey. In Pinkwart and McLaren (Eds.) 2012. *Educational Technologies for Teaching Argumentation Skills*.

[7] Green, N. 2005 A Bayesian network coding scheme for annotating biomedical information presented to genetic counseling clients. *Journal of Biomedical Informatics* 38, 130-144.

[8] Walton et al. 2008. Walton, D., C. Reed, and F. Macagno. 2008. *Argumentation Schemes*, Cambridge: Cambridge University Press.

[9] Druzdzel and Henrion 1993. Druzdzel, M. J., and Henrion, M. 1993. Efficient Reasoning in Qualitative Probabilistic Networks. In *Proceedings of the 11th National Conference on AI*, 548-553. Washington, DC.

[10] Green, N. 2010. Towards intelligent learning environments for scientific argumentation. In *Workshop on Ill-defined Problems and Ill-defined Domains, Intelligent Tutoring Systems 2010* (Pittsburgh, PA).



Problem

Give two arguments for diagnosis of that J.B. has cystic fibrosis.

Hypotheses

J. B. has cystic fibrosis.

J.B.'s brother has cystic fibrosis.

J.B.'s mother and father do not have any CFTR gene mutations.

J.B.'s mother and father are both "carriers" of the CFTR gene mutation that causes cystic fibrosis.

J.B.'s mother and father each have two mutated alleles (copies) of the CFTR gene.

Data

J. B. is a 2-year-old girl. During infancy, J.B. had diarrhea and colic. During her second year, J.B. grew poorly. On physical examination, J.B.'s weight and height plotted less than the 3rd percentile.

J. B. is a 2-year-old girl. During her second year, J.B. developed a chronic cough and had frequent upper respiratory infections.

No one else in J.B.'s family, including her mother, father, and 25-year-old brother, had poor growth, feeding disorders, or pulmonary illnesses.

Result of J.B.'s test for sweat chloride level was 75 mmol/L.

Generalizations

those that secrete mucus including the upper and lower respiratory tracts, pancreas, intestine, and sweat glands. Ten to 20 percent of cystic fibrosis patients present at birth with meconium ileus, and the remainder present with chronic respiratory complaints or poor growth, or both, later in life. The dehydrated and viscous secretions in the lungs of patients with cystic fibrosis interfere with mucociliary clearance, inhibit the function of naturally occurring antimicrobial peptides, provide a medium for growth of pathogenic organisms, and obstruct air flow. Recurrent cycles of infection, inflammation, and tissue destruction decrease the

Hypothesis 1

J. B. has cystic fibrosis.

Pro 1

Generalization 1

The CFTR gene that causes cystic fibrosis regulates the uptake of sodium and chloride from sweat as it moves through the sweat duct. In the absence of functional CFTR, the sweat has an increased sodium chloride content, and this is the basis of the historical "salty-baby syndrome" and the diagnostic sweat chloride test. In most patients with cystic fibrosis, the diagnosis can be based on the pulmonary or pancreatic findings and on an elevated level of sweat chloride (more than 60 mmol/L).

Data item 1

Result of J.B.'s test for sweat chloride level was 75 mmol/L.

.

Fig. 1. Screen shot of GAIL prototype user interface in formative evaluation of fall 2011 – spring 2012.

Formative feedback in *Digital Lofts*: Learning environments for real world innovation

Matthew W. Easterday
easterday@northwestern.edu

Daniel Rees-Lewis
daniel2011@u.northwestern.edu

Elizabeth Gerber
egerber@northwestern.edu

Northwestern University, Evanston, IL, 60208

ABSTRACT

C*ivic innovators* design real-world solutions to societal problems. Teaching civic innovation presents serious challenges in *classroom orchestration* because facilitators must manage a complex learning environment (which may include community partners, open-ended problems and long time scales) and cannot rely on traditional classroom orchestration techniques (such as fixed schedules, pre-selected topics and simplified problems). Here we consider how *digital lofts--online learning environments for civic innovation* might overcome orchestration challenges through the use of badges, cases, crowd-feedback, semi-automatically created instruction, self-assessment triggered group instruction, social media, and credentialing. Together these features create three types of feedback loops: a crowd critique loop in which learners receive formative feedback on their innovation work from a broader community, a case development loop in which examples of student work are semi-automatically created to provide instruction, and a learner-driven instructional loop, in which self-assessments determine which group instruction is provided. Researching and developing digital lofts will help us to understand how to support real-world innovation across design disciplines such as engineering, policy, writing and even science; and result in technologies for disseminating and scaling civic innovation education more broadly.

Keywords

Digital lofts, feedback, civic innovation, online learning environments

1. INTRODUCTION

Many of the challenges facing our society such as global warming, poverty, and illiteracy are *political* problems that cannot be solved through engineering alone. For example, to create environmentally sustainable cities we would have to train engineers to redesign the land, water, energy and information systems of the city. And while we do train engineers to design membrane filtration-, renewable energy-, and mass transit-systems, we do not teach them about changing economic policy to promote conservation, energy initiatives to discourage fossil fuel use, or zoning rules to encourage mass transit. We do teach engineers about complex mechanical systems and how to communicate effectively as a team, but we don't teach them that sustainable infrastructure might also require changes in policy. Even when we do teach them about policy, we don't teach them how to change it, and even if they did know how to change it, they can't change it alone, leaving us with engineers who are at the mercy of policy problems, not ones that can solve them. In short, good technology and bad policy means no impact (Easterday, 2012).

To overcome societal challenges, we must train ***civic innovators*** who can identify, design and engineer solutions to societal problems. Civic innovators must be able to develop, modify, and implement ideas while navigating ambiguous problem contexts, overcoming setbacks, and persisting through uncertainty in their community. To become civic innovators, learners must gain experience identifying and tackling complex, ill-structured design challenges that are not easily solved within a fixed time frame. Civic innovation education is thus a kind of *service learning* that *"*...integrates meaningful community service with instruction and reflection to enrich the learning experience, teach civic responsibility, and strengthen communities..." (ETR Associates, 2012). However, unlike other forms of service learning, civic innovation focuses on *design*--whereas service learning might ask students to pick up trash in a riverbed to motivate learning about ecology, civic innovation might ask students to pick up trash in a riverbed to motivate learning about ecology in order to identify, design, and engineer solutions to reduce environmental pollution.

But embedding learning in real-world activities makes civic innovation difficult to teach: individual mentoring can be effective but expensive; extra-curricular environments provide flexibility but insufficient guidance; and classroom instruction is too rigid and time-bound for solving complex societal problems. Embedding learning in real-world activities creates a serious challenge of ***classroom orchestration.*** Classroom orchestration (Dillenbourg & Jermann, 2010) involves satisfying the constraints of curriculum, assessment, time, energy, space, etc. required to promote learning in a given context. Embedding learning in the real-world increases the orchestration challenge because orchestration techniques that work in the classroom (such as using simple problems, making students complete assignments at the same pace) can't be used when learners are working on real-world problems. Adding community clients and professional design mentors only makes orchestration more challenging.

New cyberlearning technologies, such as web 2.0, social media, reputation systems, and crowdsourcing offer new ways to orchestrate learning environments for civic innovation. Just as we create instructional *lab*s to teach science, the purpose of this project is to develop instructional *lofts* to teach innovation. Our research question is: ***how might we create Digital Lofts: on-line, learning platforms for teaching civic innovation that overcome the orchestration challenge?***

Knowing how to design digital lofts that overcomes the orchestration challenge will allow us to amplify teaching resources to make civic innovation education feasible. Design principles for Digital Lofts would allow us to overcome orchestration challenge not just for civic innovation education, but for project-based learning environments as well, allowing us to design learning environments that are more sustainable, more easily scaled to new contexts, and more like real life.

2. BACKGROUND

Advantages of civic innovation learning communities

What do civic innovation learning environments look like? Civic innovation learning communities: (a) have pro-public missions, (b) teach learners how to design solutions to real problems, (c) are led by learners and supported by faculty and professional experts, and (d) extend nationally through a network of chapters. For example, in *GlobeMed,* students work on international health challenges. In *Engineers for a Sustainable World,* students work on projects that promote environmental, economic, and social sustainability. It is important to stress the pro-public mission of these learning communities. Learners are tackling problems that require them to address societal challenges and to understand policy issues. For example, by tackling the problem of energy sustainability, students are forced to consider the environmental, economic and legal policies that constrain the effectiveness of technological interventions. For this project, we consider *Design for America*, which provides an ideal model of a learning community for civic innovation.

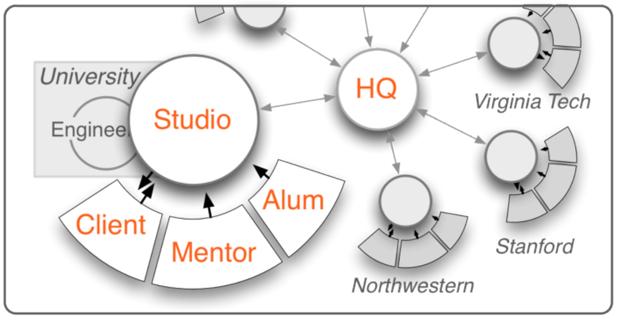


Figure 1*. Design for America's community of practice. The 14 studios are hosted on University campuses and interact with, but do not replace the existing curricula. Studios incorporate local clients, mentors and alumni and communicate directly with DFA Headquarters.*

Design for America (DFA) is a learner-directed, extracurricular service-learning environment that is succeeding at developing civic innovators. Universities host on-campus DFA studios in which student teams work on self-selected civic innovation projects throughout the academic year, applying the skills and expertise they've gained through academic coursework (Figure 1 & 2). Student teams identify challenges in healthcare, environment, and education in their local community such as reducing hospital-acquired infections and reducing water waste in cafeterias. They work with organizational partners to: understand stakeholder needs, ideate, prototype, test, and implement solutions. During the annual 4-day *Leadership Studio*, experienced student leaders train new student leaders in studio management and leadership.

Design for America was conceived by co-author Gerber during the 2008 presidential election to engage university students in solving civic issues using human-centered design. As an assistant professor of design, Gerber joined student co-founders Mert Iseri, Yuri Malina, and Hannah Chung, to start the first studio at Northwestern University. Currently, there are 14 studios hosted by universities throughout the country (including Stanford, Virginia Tech, and Northwestern) involving 1800 students (58% women), aged 18-30 from over 60 majors, working on over 50 projects; 15 faculty mentors; and 80 professional mentors. And the number of studios is expected to grow to 30 by 2015. In just four years, DFA has produced two start-ups that have raised over $1.5 million in funding. DFA has been featured in Fast Company, Oprah, and the Chicago Tribune.

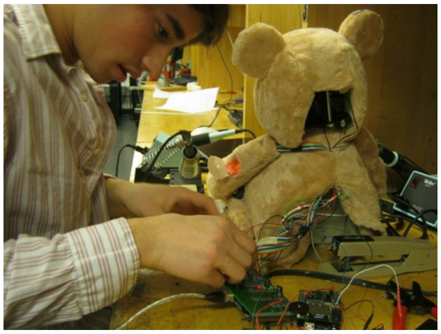


Figure 2*: Design for America students learn civic innovation through projects that require designing, building, and implementing solutions.*

Findings from surveys, daily diaries, interviews, and observations suggest that DFA students develop confidence in their ability to act as civic innovators through successful task completion, social persuasion, and vicarious learning in communities of practice with clients, peers, industry professionals, and faculty. Furthermore, students attribute achievement of learning outcomes outlined by the Accreditation Board for Engineering and Technology including identifying, formulating, and solving problems; functioning on a multidisciplinary team; communicating effectively; and knowledge of contemporary issues to their participation in Design for America. (Gerber, Marie Olson, & Komarek, 2012); (ABET Engineering Accreditation Commission, 2011).

Design for America's civic innovation model follows many recommendations of the learning sciences for improving motivation and transfer such as using real world problems that require design of meaningful products with social relevance. DFA encourages students to work on authentic problems (Shaffer & Resnick, 1999) to motivate learning and transfer. Students identify and select projects and self-direct the innovation and discovery process including observation, idea generation, prototyping, and testing (Kolodner, Crismond, Gray, Holbrook, & Puntambekar, 1998); (Puntambekar & Kolodner, 2005). By trying to apply their knowledge to a problem, students come to understand what they know and when they need more information (Edelson, 2001). Like service learning (Furco, 1996), DFA increases civic awareness, interest in the real needs of people, and contemporary issues by focusing on innovating solutions to local community challenges (Gerber et al., 2012).

Unlike traditional classrooms, Design for America's community of practice (Figure 1) expands beyond the physical boundaries of the student community to include experienced, local professionals, local clients and community members, as well as beyond the temporal boundaries of student life as learners continue to participate in projects as alumni. Students' involvement in a community of practice (Lave & Wenger, 1991) includes engaging with peer mentors, professionals and faculty in a non-evaluative environment over an extended timeframe. Communities of practice foster innovation self-efficacy (i.e., learners' belief in their ability to innovate, (Gerber et al., 2012) and such beliefs influence goal setting, effort, persistence, learning and attribution of failure (Bandura, 1997); (Deci & Ryan, 1987); (Ryan & Deci, 2000). Students select real world challenges (Shaffer & Resnick, 1999) that are personally meaningful, build and test solutions to problems, and share their work with the community through review sessions (Papert & Harel, 1991); (Papert, 1980); (Resnick, 2009); (Kolodner, Owensby, & Guzdial, 2004). Because DFA projects are extracurricular, they conclude when ideas are implemented, rather than when the academic term ends.

Orchestration challenges in civic innovation learning communities

While learning environments for civic innovation have many potential advantages, they also face many challenges. Civic innovation teachers face serious orchestration challenges because they have to teach many different project teams, with different levels of expertise, working on different problems for different community clients. The orchestration challenge makes civic innovation difficult to teach well.

Like many extra-curricular organizations, DFA students often suffer from a lack of guidance. Our needs analysis of Design for America found that, unsurprisingly, learners would benefit from more scaffolding and feedback on the innovation process including: (a) planning and conducting research on their project challenge; (b) using initial research to inform proposed solutions; (c) selecting and conducting appropriate design activities for their project challenge; and (d) discounting initial solutions if these solutions prove not to be viable. While DFA has been very successful at attracting learners, these learners report that frustrations from lack of progress makes them question their commitment to the work they are undertaking. And while leaders (student facilitators) experienced in project work and trained at the DFA leadership studio require less support, they find helping other students very challenging. In interviews, these student leaders asked for more granular 'how to' guides from DFA headquarters.

DFA students also often struggle to access available resources that could help them in their projects. While students are aware that they can reach out to experts within the DFA network generally, they struggled to identify specific individuals or instructional resources that can help them. Learners often fail to ask for support from more experienced members of the community because they don't know whom or for what to ask. Similarly, learners find it challenging to locate helpful instruction. They report floundering for long periods of time trying to find resources and as well as not knowing where to start looking.

In fact, these issues are challenges in project-based learning and criticisms of minimally guided instruction in general. Without sufficient guidance, learners become lost, confused and frustrated, which can lead to misconceptions (Kirschner, Sweller, & Clark, 2006); (Hardiman, Pollatsek, & Well, 1986); (Brown & Campione, 1996). Furthermore, students often need to develop additional help-seeking skills in order to learn effectively (Gall, 1981; Pintrich, 2004); (Ryan, Pintrich, & Midgley, 2001). Learning science provides myriad ways to offer guidance such as providing explanations, worked examples, process worksheets, prompts, (and many more) (Scardamalia, Bereiter, & Steinbach, 1984); (Reiser, 2004); (Edelson, Gordin, & Pea, 1999); (Puntambekar & Kolodner, 2005); (Kolodner et al., 2004).

Note that we do not wish to re-litigate the discovery vs. direct instructional debate here--achieving the proper balance between providing and withholding assistance (a.k.a the assistance dilemma) remains a fundamental and enduring question in the learning sciences (Koedinger & Aleven, 2007). Our point is merely that civic innovation facilitators cannot effectively deliver *any* instructional model (constructionist, direct, or otherwise) because they cannot effectively orchestrate learning at DFA studios. In other words, we cannot answer the fundamental questions about civic innovation without addressing orchestration.

The need for new orchestration technologies

In a typical classroom, orchestration is relatively easy. But the traditional classroom approaches to orchestration don't work for civic innovation. For example, to make classroom teaching easier, we often give students identical, simplified problems (in the words of one DFA student: "well-defined problems on a platter.") We use schedules that keep learners moving at the same pace so we can teach the same skills and knowledge to the whole class. This is an easy way to orchestrate groups of learners when we have a limited set of teaching resources.

Unfortunately, when we use simplified, artificial problems, we don't give students a chance to practice the skills for coping with design complexity we want them to learn. We also destroy the motivational benefits that come from working on real world problems. For example, if we want students to practice "scoping," (i.e., identifying important but tractable problems to solve) then we need to give them ill-defined problems that can be scoped in different ways and that may not fit neatly into the academic calendar. If we want them to practice communicating with clients, then we must accept unclear and changing project goals. If we want to take advantage of students' intrinsic motivation to address real world problems on topics they feel are important, then we must accept a certain level idiosyncrasy of projects. But once we start letting different groups work on different, more complex problems, at different speeds, working with clients in the community, and so on, it becomes almost impossible for a single teacher to orchestrate learning in a productive way.

Could technology help teacher orchestrate civic innovation learning environments? Existing online learning management platforms do not address the orchestration problem. Many of the most popular general-purpose online platforms assume a classroom model and are designed for distributing online books or lectures, such as academic platforms like the Open Learning Initiative (Lovett, Meyer, & Thille, 2008), MIT Opencourseware (Massachusetts Institute of Technology, 2012), and Coursera (Severance, 2012), which do not help us orchestrate design projects. Other technologies provide no pedagogical help but rather tools for managing files and conversations, such as Blackboard (Blackboard Inc., 2012), Canvas (Canvas, 2012), Lore (Lore, 2012), and Sakai (Sakai Foundation, 2012). Some technologies for orchestration focus on only small portions of the challenge such as managing a single activity (Dillenbourg & Jermann, 2010). And while there has been great progress in technologies for orchestrating scientific inquiry (Peters & Slotta, 2010), such as BioKIDS (Songer, 2006), BGuILE (Reiser et al., 2001); (Sandoval & Reiser, 2004), Inquiry Island (White et al., 2002), KIE (Bell, Davis, & Linn, 1995), and WISE (Slotta, 2004), these platforms are not appropriate for teaching civic innovation.

Solving the orchestration challenge is not simply another application of technology to teaching, it is absolutely essential for creating the civic innovation learning environments urgently needed to prepare learners for the societal challenges that await them.

3. TECHNOLOGICAL INNOVATION

Orchestration of civic innovation is difficult because there are too many moving pieces: different learners, with different abilities,

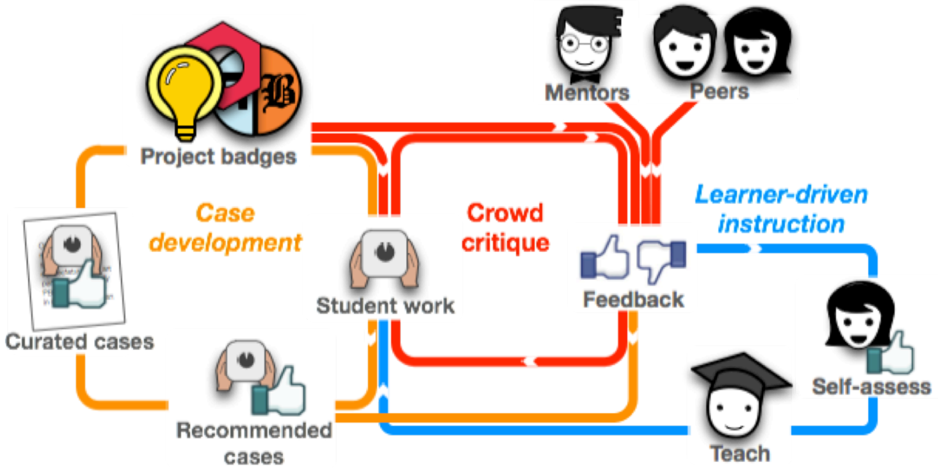


Figure 3. *Digital Lofts merge curriculum and data in three integrated feedback loops: the crowd-critique loop, the case development loop and the learner-driven instructional loop.*

working on different (complex) design problems, at different speeds, with different community clients. We could solve the orchestration challenge by giving each project team it's own professional design teacher but doing so is costly. However, with new technologies like web 2.0, crowdsourcing, and social media, we may be able to reduce the orchestration challenge for teachers and give them additional resources to overcome it. Specifically: we can use web 2.0 to scaffold the innovation process and provide flipped, just-in-time instruction relevant to students' current goals; we can use crowd-feedback to provide learners with more frequent, higher quality feedback on their progress; we can use recommender systems to semi-automatically create case libraries of successful designs; and we can automatically monitor group progress so teachers can give the right instruction to the right group at the right time.

Design hypothesis. Our initial design hypothesis argues that we can teach civic innovation by using what we call ***Digital Lofts*** to overcome the orchestration challenge. Digital Lofts are online learning platforms for support learning in real world contexts that:

1. use badges to scaffold the innovation process,
2. provide a student-generated and curated case-library linked to badges to teach design,
3. use crowd-feedback to increase the frequency and quality of feedback,
4. use recommender systems to semi-automatically create case-based instructional material,
5. use self-assessment to trigger maximally relevant group instruction,
6. use social media to facilitate participation and support, and
7. use recognition and credentialing to facilitate help-seeking and connections to resources.

These features allow us to create a curriculum that dynamically adapts to the needs of the learner, that is, to merge curriculum and data. By merging curriculum+data, we can reduce the challenge of orchestrating civic innovation to a manageable level.

To understand how Lofts help us orchestrate civic innovation, we can think of Lofts as supporting 3 interrelated feedback loops: (a) **a crowd-critique loop** in which students receive feedback on their work through project critiques, (b) a **case development loop** in which student work is used to semi-automatically create case studies of successful and unsuccessful designs which are then used to teach design principles, and (c) a **learner-driven instructional loop** in which students' self assessments trigger face-to-face group instruction taught by facilitators (Figure 3).

The crowd-critique loop

Designers and engineers often organize their work according to an innovation process. Figure 4 shows the high level steps or goals of a simplified innovation process consistent with the processes used by leading design and engineering firms like IDEO and Cooper (Dubberly, 2005) by the Stanford *d.school* (Beckman & Barry, 2007) or defined in engineering education standards (Massachusetts Department of Education, 2006). In Figure 4, the first stage of design is to "focus" by identifying a potential topic to address such as "water conservation at universities." The second stage is to "immerse" or study the user-needs, constraints and technologies involved in the issue. The third stage is to "define" a specific problem that can be solved, such as "reduce water use in the college cafeteria by 30%." The fourth stage is to "ideate" by generating a wide range of potential solutions. The fifth stage is to "build" the design using sketches, prototypes and high-fidelity implementations that realize the design idea. The sixth stage is to "test" the design. Even in simplified models like that in Figure 4, the design process is applied in an iterative and non-linear manner.

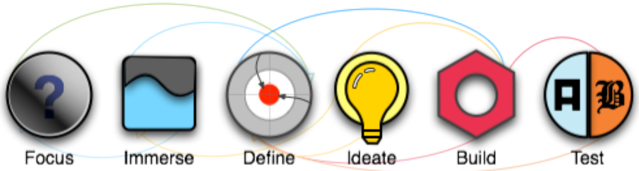


Figure 4. *Badges scaffold complex design processes for the novice into smaller, more manageable challenges and identify members who have passed the challenges as potential mentors.*

Design can be thought of as a process of learning (Beckman & Barry, 2007); (Owen, 1998). Designers construct new knowledge through observations that yield insights; insights support frameworks that inspire ideas that lead to innovative solutions (Beckman & Barry, 2007). Through this process, people construct knowledge (Dong, 2005), moving back and forth from the analytic phase of design, which focuses on finding and discovery, to the synthetic phase, which focuses on invention and making (Owen, 1998). Beckman and Barry (2007) describe knowledge creation through the design process as movement between concrete experiences and abstract conceptualization, reflective observation, and active experimentation. Inductive and deductive practices support the construction of new knowledge that designers use to shape the environment in ways that did not previously exist.

So how can teachers guide design groups working on different, complex problems? One of the most important ways to promote learning is to provide learners with scaffolding and feedback on their work.

The Loft's crowd-critique loop scaffolds the design process and provides feedback using project critiques. The crowd-critique loop starts with **project badges** (like girl scout badges) that break the complex design process into a series of manageable mini-challenges (Figure 4). For example, for the *focus* badge, learners have to scope an important but tractable issue such as *hospital acquired infections*; for the *immerse* badge, learners have to conduct user-research on their target population to better understand their needs. In the second step of the crowd-critique loop, learners use the resources attached to each badge to help them solve the challenge--each badge is linked to *flipped* (*blended)* instructional material (Khan, 2012); (Lovett et al., 2008) that includes resources, principles, and examples that can help the learners solve the design challenge. For example, the "build" badge for a web design project might include a video lecture on writing html, an interactive javascript tutorial, on-line readings about web-design principles, or examples of the different stages of creating a well-designed website In the final step of the crowd-critique loop, (after students have worked on a badge and submitted their work to the Loft), the Loft solicits feedback on students' work from professional design mentors and peers who have previously completed the badge. The mentors and peers use the badge assessment rubrics to provide feedback to students.

The widespread use of badges in online games has led to a surge of interest in badges for learning (Duncan, 2011). However, civic innovation students are already intrinsically motivated to work on real world design problems, so it doesn't make sense to use badges as extrinsic rewards that might decrease motivation (Deci, Koestner, & Ryan, 1999) and encourage gaming the system (Kraut & Resnick, 2012). So instead, Lofts use badges to scaffold the design process and communicate learning goals, which should increase learning (Ambrose, Bridges, DiPietro, Lovett, & Norman, 2010).

Combining flipped instruction with face-to-face teaching can be more effective than face-to-face teaching or online-only teaching alone (Scheines, Leinhardt, Smith, & Cho, 2005); (Lovett et al., 2008). Our flipped instructional material will use a guided-experiential learning approach shown to improve learning outcomes relative to traditional project-based learning (Velmahos et al., 2004); (Clark, 2004/2008).

Providing high quality feedback to learners is one of the most effective ways to increase learning (Hattie, 2009); (Hattie & Timperley, 2007); (Ambrose et al., 2010). The Loft provides learners with two underutilized sources of feedback: professional mentors and peers. Giving peers well-designed assessment rubrics can make their feedback as effective as instructor feedback (Sadler & Good, 2006). The Loft thus uses ***crowd-feedback*** to increase the frequency and quality of feedback available to learners.

But what if students refuse to submit work or mentors and peers refuse to review it (Kraut & Resnick, 2012)? Our needs analysis found that DFA students are hungry for feedback on their project and very willing to submit work to get this feedback. Professional design mentors are also very willing to provide this feedback assuming that students 'drive' the process by providing them with well-prepared material from their design process (which the badges help students to do).

The case development loop

Developing useful learning resources can be a challenging task especially with design teams that may all be pursuing different directions at different times--how can cyberlearning technologies help produce effective and engaging learning resources?

Our needs analysis found that DFA students prefer to share design lessons through stories about how they created their designs and how well those designs worked. In the learning sciences, this falls under the heading of *case-based reasoning,* where each story describes an example or case of a design that worked (or didn't work) along with an explanation of the key features that led the result, in which context, and so on. Teaching effectively with cases has been well studied in several forms, including learning from cases (Kolodner*,* 1993; 1997), analogies (Gentner, Loewenstein, & Thompson, 2003), and worked-examples (Ward & Sweller, 1990; Salden, Aleven, Renkl, & Schwonke, 2009).

Unfortunately, DFA students' learning from cases suffers many limitations: (a) it is done informally, so knowledge of particular cases is not spread widely; (b) students do not effectively teach with cases, sometimes hiding illustrative mistakes, promote their projects rather than teaching, and failing to highlight the key design lesson or principle; and (c) students do not present contrasting cases that would allow learners to understand the deep features and the context of applicability of a case. Such knowledge sharing is typical of large distributed organizations (Argote, 1999).

Furthermore, it is difficult to create case-based teaching material both in terms of creating a useful library of cases and in creating ways for learners to find the appropriate case when needed (Kolodner, 1997).

Digital Lofts overcome this challenge through a case development loop. In the ***case development loop***, the Loft uses assessments of students' work to semi-automatically create *case libraries*--examples of student work that include reflections about what worked, what didn't, in what context. First, the crowd-feedback from the crowd-critique loop is used to recommend particularly successful and unsuccessful examples of each design step, producing sets of contrasting cases. Second, an instructional designer creates *curated cases* by selecting cases that best illustrate key design principles. The instructional designer then refines these cases. Finally, the contrasting cases are then presented as an instructional resources linked to each badge.

The crowd-feedback and badging systems of the Loft reduce the orchestration challenge of providing relevant and engaging instruction to a manageable level in several ways. First, the Loft continually collects student work from multiple campuses, so we get the initial material for the case library "for free" using crowdsourcing, or production of work by a distributed crowd of people (Von Ahn & Dabbish, 2004). Second, project critiques act as a *recommender system* (Kiesler, Kraut, Resnick, & Kittur, 2012) sorting student work into contrasting cases. Third, cases are already linked to particular phases of the design process through the badges, so we automatically generate index that links the case to the relevant goal the student is working on. After the Digital Loft has done the heavy-lifting of generating, recommending, and indexing cases, the instructional designers can make the final case selection. Instructional designers can also edit the cases to improve their quality (Puntambekar & Kolodner, 2005; Kolodner et al., 2004), and present related so to encourage case comparison thus improving the chances of transfer (Thompson, Gentner, & Loewenstein, 2000; Gentner et al., 2003).

The learner-driven instructional loop

One of the difficulties of teaching groups of students of varying abilities engaged in projects at differing stages is how to provide face-to-face group instruction in a relevant and timely manner. When should a facilitator lead a "user research" workshop if each group is at a different stage of the design process? While the Loft tailors feedback and instruction to each project team, there is still a need for group instruction taught by a knowledgeable facilitator.

In the ***learner-driven instruction loop***, students' self assessments of their abilities and interest in learning different design skills are collected and monitored by the Loft. When enough students indicate a desire to learn a certain skill set, facilitators are notified that there is an opportunity to teach a workshop on an in-demand topic. The learner-driven instructional loop begins after students complete a badge. At this point, the Loft reminds learners to update their "individual development plans" (Beausaert, Segers, & Gijselaers, 2011). An individual development plan (IDP) is a list of skills along with the learner's self-assessment of his current ability level and desire to learn that skill. As students take on new badge challenges, the skills necessary for completing that badge are added to their IDP. Once a given number of students at a DFA studio or classroom express an interest in learning a particular skill, facilitators are notified that they should conduct a particular workshop (and provided with a facilitator's guide for that workshop). Because these workshops are triggered by students' current interests, the workshops maximally target students' interests and needs. While students may not be perfectly accurate in their self-assessments, feedback from mentors and peers provide a reality check on the students' self-assessments (i.e., negative feedback from mentors will prompt students to reassess their skills).

People who implement career goal plans report greater success and satisfaction in their career (Ng, Eby, Sorensen, & Feldman, 2005), so IDPs for civic innovation should increase the success and satisfaction of novice civic innovators on their journey to become more successful designers.

4. CONCLUSION

The study of Digital Lofts will lead empirically-grounded principles for designing online environments for civic innovation education, contributing to number of research areas including digital badges, crowdsourcing, learning-by-cases, design-based learning, and online learning communities. Because many domains can be framed as design disciplines including engineering (making technologies), policy (creating government programs), English language arts (creating texts and speeches), and even science (creating research studies), principles for online innovation education apply to myriad disciplines. And by coordinating groups of learners and mentors throughout the design process, Digital Lofts blur the boundaries between informal and formal learning environments: making extra curricular environments more effective and classroom environments more like real life. This project seeks to lay a theoretical foundation for understanding the broader ecosystem of online, social, design-based learning environments.

More broadly, our goal is to create a widely adopted online learning environment that will support civic innovation training. The Digital Loft platform will be disseminated broadly, targeting use in the teaching, training, and learning of civic innovation. This will fill an urgent need for learning environments that educate civic innovators who can solve our greatest societal challenges. Foreseeable impacts on higher education and society include: increasing the number of graduates motivated and capable of broader societal impact, improved education, curricular changes, and support for future interventions. Successful output of this project will help to foster and support a culture of innovation in our future workforce. By developing a scalable, cost-effective, online platform for design-based learning across many disciplines (design, engineering, speaking, etc.) Digital Lofts have the potential to fundamentally transform online learning.

5. ACKNOWLEDGMENTS

We are grateful to the participants of Design for America for their enthusiastic participation. This work was generously funded by the Digital Medial Learning Competition supported by the MacArthur and Mozilla Foundations.

REFERENCES

ABET Engineering Accreditation Commission. (2011). *Criteria for accrediting engineering programs*. Baltimore, MD: ABET. Retrieved from http://www.abet.org/uploadedFiles/Accreditation/Accreditation_Process/Accreditation_Documents/Current/eac-criteria-2012-2013.pdf

Ambrose, S. A., Bridges, M. W., DiPietro, M., Lovett, M. C., & Norman, M. K. (2010). *How learning works: 7 research-based principles for smart teaching.* San Francisco, CA: Jossey-Bass.

Argote, L. (1999). *Organizational learning: Creating, retaining and transferring knowledge.* New York: Springer.

Bandura, A. (1997). *Self-efficacy: The exercise of control.* W. H. Freeman and Company.

Beausaert, S., Segers, M., & Gijselaers, W. (2011). The use of a personal development plan and the undertaking of learning activities, expertise-growth, flexibility and performance: The role of supporting assessment conditions. *Human Resource Development International*, *14*(5), 527-543.

Beckman, S. L., & Barry, M. (2007). Innovation as a learning process: Embedding design thinking. *California Management Review*, *50*(1), 25.

Bell, P., Davis, E. A., & Linn, M. C. (1995). The knowledge integration environment: Theory and design. In *The first international conference on computer support for collaborative learning* (pp. 14-21).

Blackboard Inc. (2012). *Blackboard* [Computer Software]. Retrieved from http://www.blackboard.com/

Brown, A. L., & Campione, J. C. (1996). Psychological theory and the design of innovative learning environments: On procedures, principles, and systems. In L. Schauble & R. Glaser (Eds.), *Innovations in learning: New environments for education* (pp. 289-322). Mahway, NJ: Erlbaum.

Canvas. (2012). *Instructure* [Computer Software]. Retrieved from http://www.instructure.com/

Clark, R. E. (2008). *Design document for A guided experiential learning course*. Submitted to satisfy contract DAAD 19-99-D-0046-0004 from TRADOC to the Institute for Creative Technologies and the Rossier School of Education, University of Southern California. (Original work published 2004) Retrieved from http://www.cogtech.usc.edu/publications/gel_design_document.pdf

Deci, E. L., & Ryan. (1987). The support of autonomy and the control of behavior. *Journal of Personality and Social Psychology*, 53(6), 1024-1037

Deci, E. L., Koestner, R., & Ryan, R. M. (1999). A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation. *Psychological Bulletin*, *125*(6), 627-668.

Dillenbourg, P., & Jermann, P. (2010). Technology for classroom orchestration. In M. S. Khine & I. M. Salhey (Eds.), *New science of learning* (pp. 525-552). New York: Springer.

Dong, A. (2005). The latent semantic approach to studying design team communication. *Design Studies*, *26*(5), 445-461.

Dubberly, H. (2005). *How Do You Design?* Retrieved from http://www.dubberly.com/wp-content/uploads/2008/06/ddo_designprocess.pdf

Duncan, A. (2011, September 15). Digital badges for learning: Remarks by secretary duncan at 4th annual launch of the macarthur foundation digital media and lifelong learning competition. Retrieved from http://www.ed.gov/news/speeches/digital-badges-learning

Easterday, M. W. (2012). Matthew easterday: Cyber-Civics 101. Presentation at the NSF cyberlearning summit, jan 18, 2012, washington D.C. Retrieved from from http://www.youtube.com/watch?v=UBPeDVR2nOo&feature=youtu.be

Edelson, D. C. (2001). Learning-for-Use: A framework for the design of technology-supported inquiry activities . *Journal of Research in Science Teaching*, *38*(3), 355-385.

Edelson, D. C., Gordin, D. N., & Pea, R. D. (1999). Addressing the challenges of inquiry-based learning through technology and curriculum design. *Journal of the Learning Sciences*, *8*(3-4), 391-450.

ETR Associates. (2012). What is service-learning? [Web page] Retrieved from http://www.servicelearning.org/what-service-learning.

Furco, A. (1996). Service-learning: A balanced approach to experiential education. *Expanding Boundaries: Serving and Learning*, *1*, 1-6.

Gall, S. N. -L. (1981). Help-seeking: An understudied problem-solving skill in children. *Developmental Review*, *1*(3), 224 - 246. doi:10.1016/0273-2297(81)90019-8

Gentner, D., Loewenstein, J., & Thompson, L. (2003). Learning and transfer: A general role for analogical encoding. *Journal of Educational Psychology*, *95*(2), 393.

Gerber, E. M., Marie Olson, J., & Komarek, R. L. D. (2012). Extracurricular design-based learning: Preparing students for careers in innovation. *International Journal of Engineering Education*, *28*(2), 317.

Hardiman, P. T., Pollatsek, A., & Well, A. D. (1986). Learning to understand the balance beam. *Cognition and Instruction*, *3*(1), 63-86.

Hattie, J. (2009). *Visible learning: A synthesis of over 800 meta-analyses relating to achievement.*

Hattie, J., & Timperley, H. (2007). The power of feedback. *Review of Educational Research*, *77*(1), 81-112.

Khan, S. (2012). *The khan academy.* [Web page] Retrieved from http://www.khanacademy.org/

Kiesler, S., Kraut, R., Resnick, P., & Kittur, A. (2012). Regulating behavior in online communities. In P. Resnick & R. Kraut (Eds.), *Evidence-based social design: Mining the social sciences to build online communities* (pp. 125-178). Cambridge, MA: MIT Press.

Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist*, *41*(2), 75-86.

Koedinger, K. R., & Aleven, V. (2007). Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review*, *19*(3), 239-264.

Kolodner, J. L. (ed.) (1993). *Case-based learning.* Boston: Springer.

Kolodner, J. L. (1997). Educational implications of analogy: A view from case-based reasoning. *American Psychologis*, *52*(1), 57.

Kolodner, J. L., Crismond, D., Gray, J., Holbrook, J., & Puntambekar, S. (1998). Learning by design from theory to practice. *Proceedings of the International Conference of the Learning Sciences*, 16-22.

Kolodner, J. L., Owensby, J. N., & Guzdial, M. (2004). Case-based learning aids. In *Handbook of research for educational communications and technology* (Vol. 2, pp. 829-861).

Kraut, R., & Resnick, P. (2012). Encouraging contribution to online communities. In P. Resnick & R. Kraut (Eds.), *Evidence-based social design: Mining the social sciences to build online communities* (pp. 21-76). Cambridge, MA: MIT Press.

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation.* Cambridge university press.

Lore. (2012). Lore. [Computer Software] Retrieved from http://lore.com/

Lovett, M., Meyer, O., & Thille, C. (2008). The open learning initiative: Measuring the effectiveness of the OLI statistics course in accelerating student learning. *Journal of Interactive Media in Education.* Retrieved from *http://jime.open.ac.uk/2008/14*

*M*assachusetts Department of Education. (2006). *Massachusetts science and technology/engineering curriculum framework*. Massachusetts. Retrieved from www.doe.mass.edu/frameworks/scitech/1006.pdf

Massachusetts Institute of Technology. (2012). *MIT opencourseware* [Computer Software]. Retrieved from http://ocw.mit.edu/index.htm

Ng, T. W. H., Eby, L. T., Sorensen, K. L., & Feldman, D. C. (2005). Predictors of objective and subjective career success: A meta-analysis. *Personnel Psychology*, *58*(2), 367-408.

Owen, C. L. (1998). Design research: Building the knowledge base. *Design Studies*, *19*(1), 9-20.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.

Papert, S., & Harel, I. (1991). Situating constructionism. In *Constructionism* (pp. 1-11).

Peters, V. L., & Slotta, J. D. (2010). Scaffolding knowledge communities in the classroom: New opportunities in the web 2.0 era. In *Designs for learning environments of the future* (pp. 205-232). New York: Springer.

Pintrich, P. R. (2004). A conceptual framework for assessing motivation and self-regulated learning in college students. *Educational Psychology Review*, *16*(4), 385-407.

Puntambekar, S., & Kolodner, J. L. (2005). Toward implementing distributed scaffolding: Helping students learn science from design. *Journal of Research in Science Teaching*, *42*(2), 185-217.

Reiser, B. J. (2004). Scaffolding complex learning: The mechanisms of structuring and problematizing student work. *The Journal of the Learning Sciences*, *13*(3), 273-304.

Reiser, B. J., Tabak, I., Sandoval, W. A., Smith, B. K., Steinmuller, F., & Leone, A. J. (2001). BGuILE: Strategic and conceptual scaffolds for scientific inquiry in biology classrooms. In *Cognition and instruction* (Vol. 25, pp. 263-306).

Resnick, M. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60-67. doi:10.1145/1592761.159277

Ryan, A. M., Pintrich, P. R., & Midgley, C. (2001). Avoiding seeking help in the classroom: Who and why? *Educational Psychology Review*, *13*(2), 93-114.

Ryan, R. M., & Deci, E. L. (2000). Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology*, *25*(1), 54-67. doi:10.1006/ceps.1999.1020

Sadler, P. M., & Good, E. (2006). The impact of self-and peer-grading on student learning. *Educational Assessment*, *11*(1), 1-31.

Sakai Foundation. (2012). *Sakai* [Computer Software]. Retrieved from http://www.sakaiproject.org/

Salden, R. J. C. M., Aleven, V. A. W. M. M., Renkl, A., & Schwonke, R. (2009). Worked examples and tutored problem solving: Redundant or synergistic forms of support? *Topics in Cognitive Science*, *1*(1), 203-213. doi:10.1111/j.1756-8765.2008.01011.x

Sandoval, W. A., & Reiser, B. J. (2004). Explanation-driven inquiry: Integrating conceptual and epistemic scaffolds for scientific inquiry. *Science Education*, *88*(3), 345-372.

Scardamalia, M., Bereiter, C., & Steinbach, R. (1984). Teachability of reflective processes in written composition. *Cognitive Science*, *8*(2), 173-190.

Scheines, R., Leinhardt, G., Smith, J., & Cho, K. (2005). Replacing lecture with web-based course materials. *Journal of Educational Computing Research*, *32*(1), 1-26.

Severance, C. (2012). Teaching the world: Daphne koller and coursera. *Computer*, *45*(8), 8-9.

Shaffer, D. W., & Resnick, M. (1999). Thick" authenticity: New media and authentic learning. *Journal of Interactive Learning Research*, *10*(2), 195-215.

Slotta, J. D. (2004). The web-based inquiry science environment (WISE): Scaffolding knowledge integration in the science classroom. In M. C. Linn, E. A. Davis, & P. Bell (Eds.), *Internet environments for science education* (pp. 203-232). Lawrence Erlbaum Mahwah, NJ.

Songer, N. B. (2006). BioKIDS: An animated conversation on the development of complex reasoning in science. In R. Keith Sawyer (Ed.), *The cambridge handbook of the learning sciences* (pp. 355-370). New York: Cambridge University Press.

Thompson, L., Gentner, D., & Loewenstein, J. (2000). Avoiding missed opportunities in managerial life: Analogical training more powerful than individual case training. *Organizational Behavior and Human Decision Processes*, *82*(1), 60-75.

Velmahos, G. C., Toutouzas, K. G., Sillin, L. F., Chan, L., Clark, R. E., Theodorou, D., & Maupin, F. (2004). Cognitive task analysis for teaching technical skills in an inanimate surgical skills laboratory. *American Journal of Surgery*, *187*(1), 114-9.

Von Ahn, L., & Dabbish, L. (2004). Labeling images with a computer game. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 319-326).

Ward, M., & Sweller, J. (1990). Structuring effective worked examples. *Cognition and Instruction*, *7*(1), 1-39.

White, B., Frederiksen, J., Frederiksen, T., Eslinger, E., Loper, S., & Collins, A. (2002). Inquiry island: Affordances of a multi-agent environment for scientific inquiry and reflective learning. In *Proceedings of the fifth international conference of the learning sciences (ICLS)* . Mahwah, NJ: Erlbaum.

What is my essay really saying? Using extractive summarization to motivate reflection and redrafting

Nicolas Van Labeke[1], Denise Whitelock[1], Debora Field[2], Stephen Pulman[2], John Richardson[1]

[1] Institute of Educational Technology
The Open University
Walton Hall, Milton Keynes, MK7 6AA, UK
Nicolas.Vanlabeke@open.ac.uk
Denise.Whitelock@open.ac.uk
John.T.E.Richardson@open.ac.uk

[2] Department of Computer Science
University of Oxford
Wolfson Building, Parks Road, Oxford OX1 3QD, UK
stephen.pulman@cs.ox.ac.uk
debora.field@cs.ox.ac.uk

ABSTRACT

This paper reports on progress on the design of OpenEssayist, a web application that aims at supporting students in writing essays. The system uses techniques from Natural Language Processing to automatically extract summaries from free-text essays, such as key words and key sentences, and carries out essay structure recognition. The current design approach described in this paper has led to a more "explore and discover" environment, where several external representations of these summarization elements would be presented to students, allowing them to freely explore the feedback, discover issues that might have been overlooked and reflect on their writing. Proposals for more interactive, reflective activities to structure such exploration are currently being tested.

Keywords

Essay writing; Extractive Summarization; Formative Feedback; External Representations; Reflective Activities.

1. INTRODUCTION

Written discourse is a major class of data that learners produce in online environments, arguably the primary class of data that can give us insights into deeper learning and higher order qualities such as critical thinking, argumentation and mastery of complex ideas. These skills are indeed difficult to master as illustrated in the revision of Bloom's Taxonomy of Educational Objectives (Pickard 2007) and are a distinct requirement for assessment in higher education. Assessment is an important component of learning and in fact (Rowntree 1987) argues that it is the main driver for learning and so the challenge is to provide an effective automated interactive feedback system that yields an acceptable level of support for university students writing essays.

Effective feedback requires that students are assisted to manage their current essay-writing tasks and to support the development of their essay-writing skills through effective self-regulation.

Our research involves using state-of-the-art techniques for analyzing essays and developing a set of feedback models which will initiate a set of reflective dialogic practices. The main pedagogical thrust of e-Assessment of free-text projects is how to provide meaningful "advice for action" (Whitelock 2010) in order to support students writing their summative assessments. It is the combination of incisive learning analytics and meaningful feedback to students which is central to the planning of our empirical studies. Specifically, we are investigating whether summarization techniques (Lloret & Palomar 2012) could be used to generate formative feedback on free-text essays submitted by students.

This paper is organized as follows. We briefly describe the context and research questions that are informing the design principles of our platform, OpenEssayist. We then describe the basic processes behind the summarization techniques implemented in the system and, finally, demonstrate the current stage of design of the prototype, in particular the use of external representations for the summarization elements. We conclude this paper by sketching our current and planned evaluations.

2. DEFINING A DESIGN SPACE FOR OPENESSAYIST

2.1 WRITING SUMMARIES VS. REFLECTING ON SUMMARIES FOR WRITING.

Writing summaries has been a long-standing educational activity and has received some serious attention in delivering computer-based support. For example, systems such as SummaryStreet (Wade-Stein & Kintsch 2004) or Pensum (Villiot-Leclercq *et al.* 2010) aim to help students *write* summaries as a learning, skills-based, task.

But using summaries as a source of reflection on your own writing seems to be a more open issue. Recent research on formative feedback suggests indeed that essay summarization, understood to comprise both a short summary of the essay and a simple list of its main topics, could be useful for students, e.g. "*to help determine whether the actual performance was the same as the intended performance*" (Nelson & Schunn 2009, p. 378).

With this in mind, one of our research questions is how to use advances in Natural Language Processing to design an automated summarization engine that would provide a good foundation for a dedicated model of formative feedback. Can we use summarization elements to help students identify or visualize patterns in their essays, as explored by (O'Rourke & Calvo 2009)? Or to trigger questions and reflective activities, as implemented in Glosser (Villalon *et al.* 2008)?

2.2 SUPPORTING ESSAY WRITING IN DISTANCE LEARNING

The context of application of our research agenda is supporting students at the Open University (OU) in writing assignment essays. Specifically, we have been working closely with a postgraduate module *Accessible online learning: Supporting disabled students* (referred to as H810). This postgraduate module runs twice a year for about 20 weeks and contributes to a Master of Arts (MA) in Online and Distance Education. All courses, materials and support are delivered online. Students on this module, as is the case for most of the students at the OU, are typically part-time, mature students, who have not been in formal education for a long period of time. It is therefore unsurprising that writing essays, a common assignment in most of the OU courses, proves to be a challenging task for students (and, anecdotal evidence suggests, a common reason for drop-out).

At the same time, OU students often have extensive work experience in a wide variety of areas, and that experience is explicitly capitalized on in the assignments. This means that essays can vary greatly in subject matter. To illustrate this point, two examples of assignment tasks are given in Table 1.

Table 1. Examples of assignment tasks.

TMA1 (1500 words)
Write a report explaining the main accessibility challenges for disabled learners that you work with or support in your own work context(s). Use examples from your own experience, supported by the research and practice literature. If you're not a practitioner, write from the perspective of a person in a relevant context. Critically evaluate the influence of the context (e.g. country, institution, perceived role of online learning within education) on the: (1) identified challenges; (2) influence of legislation; (3) roles and responsibilities of key individuals; (4) role of assistive technologies in addressing these challenges.
TMA2 (3000 words)
Critically evaluate your own learning resource in the following ways: (1) Briefly describe the resource and its accessibility features; (2) Evaluate the accessibility of your resource, identifying its strengths and weaknesses; (3) Reflect on the processes of creating and evaluating accessible resources.

The questions we are considering, given this context, is how we can support these students as they write essays and what the implications are for the design of a computer- and summarization-based approach.

In the initial phase of the project, we ran a couple of focus groups with OU students that helped to identify many aspects of the students' personal approach to essay writing (Alden *et al.* 2013).

Writing an essay is a task that can involve several stages: preparation of material, drafting of essay, reflecting on feedback, summative evaluation by tutors. But not all of them are suitable, or even desirable, for support in an automated assessment system.

Moreover, writing a 1500+ word essay is not a casual operation, nor is it handled in the same way by different students. For example, we discovered that some students are not using computers to draft their essays, because of unease, lack of permanent access to a desktop computer or simply because they still prefer to write their text with paper-and-pencil before typing for the final submission.

Relying on embedded text editors or on cloud-based solutions such as Google Docs – as done by (Southavilay *et al.* 2013) for collaborative writing – is therefore not a viable solution in our context. The system will have to accept texts written with whatever platform students are using to organize, draft and revise their essay. Ultimately, the system will have to be seen and used as a resource, the way forums, online textbooks and other digital tools are used by OU students.

One of the consequences of such selective support is that the flow of activities during the overall writing process is likely to be highly scattered in time: the core of the activity (i.e. writing) will take place *outside* the system's ecology and its use will be mostly as an ancillary to that main task. Careful attention will have to be paid to trade-offs between support and distraction, especially when it comes to interaction, formal reflective activities, accessibility and usability[1].

Finally, the diversity of content in student essays is one of the motivations for investigating summarization techniques as a backbone for formative feedback. Unlike other NLP techniques such as Latent Semantic Analysis (LSA), used in many educational systems, we will not be relying on a corpus of essays to compare and grade new essays accordingly. Summarization using the text alone with no domain-specific knowledge will enable OpenEssayist to handle assignments which have open topics, as well as enabling it to be applied without extensive further development to new subject areas.

2.3 A WEB APPLICATION FOR SUMMARIZATION-BASED FORMATIVE FEEDBACK.

OpenEssayist is developed as a web application and is composed primarily of two components (Figure 1, see appendix). The first component, EssayAnalyser, is the summarization engine, implemented in Python with NLTK[2] (Bird *et al.* 2009) and other toolkits. It is being designed as a stand-alone RESTful web service, delivering the basic summarization techniques that will be consumed by the main system. The second component is OpenEssayist itself, implemented on a PHP framework. The core system consists of the operational back-end (user identification, database management, service brokers, feedback orchestrator) and the cross-platform, responsive HTML5 front-end.

The intended flow of activities within the system can be summarized as follows. Students are registered users and have assignments, defined by academic staff, allocated to them. Once they have prepared a draft offline and seek to obtain feedback, they log on to the OpenEssayist system and submit their essay for analysis, either by copy-and-paste or by uploading their document. OpenEssayist submits the raw text to the EssayAnalyser service and, upon completion, retrieves and stores the summarization data. From that point on, the students, at their own pace, can then explore the data using various external

[1] Worth noting is that students who mention that they don't use computers for drafting their essays also report that they are using smart phones. A focus on responsive user interface suitable for mobile (and tablet) and on asynchronous data access will be an issue for serious consideration in this project.

[2] Natural Language Processing Toolkit, see http://nltk.org/

representations made available to them, can follow the prompts and trigger questions that the Feedback Orchestrator might generate from the analysis and can then start planning their next draft accordingly.

Again, this rewriting phase will take place offline, the system merely offering repeated access to the summarization data and feedback, as a resource, until the students are prepared to submit and explore the summarization feedback on their second draft and on the changes across drafts. This cycle of submission, analysis and revision continues until the students consider their essay ready for summative assessment.

3. EXTRACTIVE SUMMARIZATION

We decided to start experimenting with two simpler summarization strategies that could be implemented fairly quickly: key phrase extraction and extractive summarization, following the TextRank approach proposed and evaluated in (Mihalcea & Tarau 2004). Key phrase extraction aims at identifying which individual words or short phrases are the most suggestive of the content of a discourse, while extractive summarization is essentially the identification of whole key sentences. Our hypothesis is that the quality and position of key phrases and key sentences within an essay (i.e., relative to the position of its structural components) might give an idea of how complete and well-structured the essay is, and therefore provide a basis for building suitable models of feedback.

The implementation of these summarization techniques is based on three main automatic processes: 1) recognition of essay structure; 2) unsupervised extraction of key words and phrases; 3) unsupervised extraction of key sentences.

Before extracting key terms and sentences from the text, the text is automatically pre-processed using some of the NLTK modules (tokenizer, lemmatizer, part-of-speech tagger, list of stop words).

3.1 STRUCTURE IDENTIFICATION

The automatic identification of essay structure is carried out using handcrafted rules developed through experimentation with a corpus of 135 essays that have been previously submitted for the same H810 module. The system tries to automatically recognize which structural role is played by each paragraph in the essay (summary, introduction, conclusion, discussion, references, etc.). This identification is achieved regardless of the presence of content-specific headings and without getting clues from formatting mark-up. With the essays in the corpus varying greatly in structure and formatting, it was decided that structure recognition would be best achieved without referring to a high-level formatting mark-up.

3.2 KEY WORD EXTRACTION

EssayAnalyser uses graph-based ranking methods to perform unsupervised extractive summarization of key words. The 'key-ness' value of a word can be understood as its 'significance within the context of the overall text'.

To compute this key-ness value, each unique word in the essay is represented by a node in a graph, and co-occurrence relations (specifically, within-sentence word adjacency) are represented by edges in the graph. A centrality algorithm – we have experimented with betweenness centrality (Freeman 1977) and PageRank (Brin & Page 1998) – is used to calculate the significance of each word. Roughly speaking, a word with a high centrality score is a word that sits adjacent to many other unique words which sit adjacent to many other unique words which…, and so on. The words with high centrality scores are the key words[3].

Since a centrality score is attributed to *every* unique word in the essay, a decision needs to be made as to what proportion of the essay's unique words qualify as key words. The distribution of key word scores follows the same shape for all essays, an acute "elbow" and then a very long tail, observed for word adjacency graphs by (Ferrer i Cancho & Solé 2001). We therefore currently take the key-ness threshold to be the place where the elbow bend appears to be sharpest.

Once key words have been identified, the system matches sequences of these against the surface text to identify within-sentence key phrases (bigrams, trigrams and quadgrams).

3.3 KEY SENTENCE EXTRACTION

A similar graph-based ranking approach is used to compute key-ness scores to rank the essay's sentences. Instead of word adjacency (as in the key word graph), co-occurrence of words across pairs of sentences is the relation used to construct the graph. More specifically, we currently use cosine similarity to derive a similarity score for each pair of sentences. Whole sentences become nodes in the graph, while the similarity scores become weights on the edges connecting pairs of sentences. The TextRank key sentence algorithm is then applied to the graph to compute the centrality scores.

3.4 ESSAY ANALYSIS OUTPUT

The text submitted for analysis is stripped of its surface formatting and returned as a *new* annotated structured text, reflecting the various elements identified by EssayAnalyser: sentences and paragraphs, labeled with their structural roles (body, introduction, headings, conclusions, captions, etc.) and confidence levels.

Key words and key phrases are returned as an ordered list of terms, associated with various metrics such as centrality, frequency of inflected forms, etc. Key sentences are identified within the annotated text by their ranked centrality scores.

In addition to the core summaries of the essay, various metrics and specialized data structures are made available, for use by the system for diagnosis purpose (or by researchers for analysis): word and sentence graphs, word count, paragraph and sentence density and length, number of words in common with the module textbook, average frequency of the top handful of most frequent words, etc.

Our task is now to look for ways of presenting and exploiting these results and, ultimately, to devise effective models of feedback using them.

4. OPENESSAYIST: EXTERNAL REPRESENTATIONS AND REFLECTIVE ACTIVITIES

The design of the first version of the system has focused on defining the essay summarization engine and integrating it into a working web application that supports draft submission, analysis and reporting, using multiple external representations.

[3] In the actual process, we are in fact ranking *lemmas* (the canonical form of a set of words) rather than their inflected forms in the surface text. For brevity’s sake, we will keep the terms ‘words’ and ‘key words’ in this document.

At the front-end level, the instructional interactions have been deliberately limited to fairly unconstrained forms, leading the system towards a more "explore and discover" environment. Our aim was to establish a space where emerging properties of the interventions under investigation (i.e. using summarization techniques for generating formative feedback) could be discovered, explored and integrated into the design cycles in a systematic way, contributing to both the end-product of the design cycle (the system itself) and to its theoretical foundations.

Several external representations have been designed and deployed in the system, reporting the different elements described above in different ways, trying to highlight such properties in the current essay (or, in changes over successive drafts).

The main view of the system is a mash-up of the re-structured raw text, highlighting many of the features extracted by EssayAnalyser in context, using a combination of HTML markers and JavaScript-enabled interactive displays (Figure 2). Sentences, paragraphs and headings (as identified by EssayAnalyser) are displayed as blocks of text, with visual markers on the left-hand side indicating their diagnosed structural role (e.g. introduction, headings, conclusion, etc.). Key words and key phrases are also highlighted with specific visual markers, as with the top-ranked key sentences.

A control-box allows the student to change the visibility of selected elements of the essay: show/hide specific structural components (e.g. only show the introduction), key words (or user-defined categories, see below), top-ranked sentences, etc. (Figure 3).

The intended purpose of this dynamic essay representation is to attract the attention of the student away from the surface text to issues at a more structural level that might become apparent once an alternative viewpoint is considered.

For example, if confidence levels were low in the structural recognition of an introduction, the visual indicator would reflects that degree of (un)certainty about their exact role of this paragraph, requiring the student to reflect on his intention (or on the fact that an introduction might be missing in the essay or seems to be too long or too short).

Similarly, the highlighting of key words and key phrases, in context within the essay, is intended to trigger reflection on their occurrence within the text. Its purpose is different from a dedicated external representation of the key words as such (Figure 4), where the focus is more on individual terms, and on their relative importance in the essay (as indicated by their centrality score or frequency in the surface text). In the mash-up view, the key word centrality score is played down (we do not represent any attribute other than its identification as a key word) while we try to focus on whether key word *dispersion* across the essay might help identify the flow of ideas and arguments.

To complement the main mash-up view and to alleviate potential overload, we are also designing and deploying ad-hoc external representations on specific aspects of the summarization.

For example, we are exploring whether more compact representations of the dispersion of key words across the essay (Figure 5) might provide a more suitable ground for insight into its meaning. In this graph, each key word (or category of key words, if they have been defined) is plotted on a scale showing the flow of the essay (the figure uses words on the x-axis but sentences and paragraphs can also be used as units). By adding on the scale markers for the introduction, the conclusion (or any other structural elements), the student has immediate access to the overall flow of key words across the text and within specific parts of it: patterns of occurrence or omission might provide opportunity to detect an overlooked mistake (e.g. what can be said about the fact that "learning resource", ranked as a top key word by the system, only occurs in the first few paragraphs of the essay?).

On a more experimental approach, we are also exploring the possibility of visually exploiting the networks that constitute the core internal representation of the key word and key sentence extraction, using various visualization tools (e.g. force-directed graph, adjacency matrix). A case for their informational and – more importantly – formative values remains to be made.

However, we are also arguing that, to help students explore the significance of summarization elements in their essay, visualization on its own will not be enough. Support for reflective *action* is needed to resolve a key question students are likely to ask: "what are the key words (and key sentences) and how do they help me?"

Let's consider the key words. In the current version of the system, key words are presented in a very simple fashion (Figure 4): ranked by their centrality score and by their dimension (i.e. bigrams, trigrams and so on). This is a reflection of the domain-independent, data-driven design approach followed so far; key words are derived on the basis of co-occurrence, i.e. identity relation, not on the basis of semantic relations such as synonymy or hyponymy.

We can therefore have situations, as in Figure 4, where key words such as "learning experience" and "study experience" both occur as distinct bigrams, whereas, for the student who used them, they might mean very similar things. More fine-grained approaches could be implemented in EssayAnalyser to address such situation at detection level, but, ultimately, the *intention* of the student is the only safe ground for deciding on the usage of both terms. Hence the need to support some user interaction with the system, especially if it can act as a reflective scaffold.

A first example of support for reflective action is made available to the students immediately after a draft has been analyzed by the system: to let them organize key words according to their own schema, using as many categories as they wish or need (see Figure 6). This serves two purposes: it helps the students to reflect on the content of the essay and helps the system to adapt the content of every external representation accordingly, by clustering key words together (as seen in Figure 5).

Another key-word-related activity relies on the fact that a decision is made by the system on what constitutes a key word, a decision that might be at odds with the intention of the student. So we are offering the possibility for students to define – or select – their own key words. With the extraction process deriving a centrality score and frequency count for every unique word in the text, the student's decision to flag a word as a key word can be matched with that information, encouraging her to reflect on why it might be that the words she thinks should be key words are not being recognized by the system as such.

5. CONCLUSION

The first phase of the design of OpenEssayist, as reported in this paper, has focused on devising a range of external representations on the various elements that the summarization engine is extracting, notably key words, key sentences and the structural role of paragraphs in the essay.

We have implemented a working prototype that delivers a fairly unconstrained, unstructured exploration of these elements, The drive of our design approach has been to consider how these elements, either separately or combined, would create a space where students (and researchers) could discover emerging properties of the essay, triggering deeper reflection on their own writing.

Our objective is now to consider how we structure these reflective episodes for support within the system, and how we design dedicated reflective activities that will prove to deliver formative feedback for students.

Our work is continuously focusing on three parallel but inter-connected lines of experimentation and evaluation:

1) improve the different aspects of the summarization engine;
2) experiment with it on various corpora of essays to identify trends and markers that could be used as progress and/or performance indicators (Field *et al*. 2013);
3) refine the educational aspect of the system, identify possible usage scenarios (Alden *et al*. 2013), test pedagogical hypotheses and models of feedback.

At the time of writing, several usability/desirability inspection sessions are underway, using both semi-structured walkthrough protocols in a usability lab and self-guided remote sessions with students from the last presentation of the H810 module. Part of the aim of these empirical studies is to identify tutorial strategies to be used to scaffold the student's exploitation of the system.

Finally, we are planning two empirical educational evaluations of OpenEssayist in an authentic e-learning context, to take place in September 2013 and February 2014. All students enrolled on two different Master's degree modules will be offered access to the system for two of the module's assignments and encouraged to submit multiple drafts of their essays. In-system data collection, post-module surveys, and interviews with selected participants and their tutors will give us valuable information on their learning experience with the system.

ACKNOWLEDGEMENTS

This work is supported by the Engineering and Physical Sciences Research Council (EPSRC, grant numbers EP/J005959/1 & EP/J005231/1).

REFERENCES

Alden, B., Van Labeke, N., Field, D., Pulman, S., Richardson, J. T. E., and Whitelock, D. (2013). Using student experience to inform the design of an automated feedback system for essay answers. In *Proceedings of the 2013 International Computer Assisted Assessment Conference* (CAA'13, Southampton, UK). pp. to appear.

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. Cambridge, MA: O'Reilly Media, Inc.

Brin, S., and Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems* 30(1), pp. 107–117.

Ferrer i Cancho, R., and Solé, R. V. (2001). The small world of human language. *Proceedings of the Royal Society of London. Series B: Biological Sciences* 268(1482), pp. 2261–2265.

Field, D., Richardson, J. T. E., Pulman, S., Van Labeke, N., and Whitelock, D. (2013). Reflections on characteristics of university students' essays through experimentation with domain-independent natural language processing techniques. In *Proceedings of the 2013 International Computer Assisted Assessment Conference* (CAA'13, Southampton, UK). pp. to appear.

Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry* 40(1), pp. 35–41.

Lloret, E., and Palomar, M. (2012). Text summarisation in progress: a literature review. *Artificial Intelligence Review* 37(1), pp. 1–41.

Mihalcea, R., and Tarau, P. (2004). TextRank: Bringing order into texts. In *Proceedings of Empirical Methods in Natural Language Processing* (EMNLP'04, Barcelona, Spain). , pp. 404–411.

Nelson, M. M., and Schunn, C. D. (2009). The nature of feedback: how different types of peer feedback affect writing performance. *Instructional Science* 37(4), pp. 375–401.

O'Rourke, S. T., and Calvo, R. A. (2009). Analysing Semantic Flow in Academic Writing. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education* (AIED'09, Brighton, UK). IOS Press, pp. 173–180.

Pickard, M. J. (2007). The new Bloom's taxonomy: An overview for family and consumer sciences. *Journal of Family and Consumer Sciences Education* 25(1), pp. 45–55.

Rowntree, D. (1987). *Assessing Students: How Shall We Know Them?* London: Kogan Page.

Southavilay, V., Yacef, K., Reimann, P., and Calvo, R. A. (2013). Analysis of Collaborative Writing Processes Using Revision Maps and Probabilistic Topic Models. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge* (LAK'13, Leuven, Belgium). ACM, pp. 38–47.

Villalon, J., Kearney, P., Calvo, R. A., and Reimann, P. (2008). Glosser: Enhanced Feedback for Student Writing Tasks. In *Proceedings of the 8th IEEE International Conference on Advanced Learning Technologies* (ICALT'08, Santander, Spain). IEEE Press, pp. 454–458.

Villiot-Leclercq, E., Mandin, S., Dessus, P., and Zampa, V. (2010). Helping Students Understand Courses through Written Syntheses: An LSA-Based Online Advisor. In *Proceedings of the 10th International Conference on Advanced Learning Technologies (ICALT)* (ICALT'10, Sousse, Tunisia). IEEE Press, pp. 341–343.

Wade-Stein, D., and Kintsch, E. (2004). Summary Street: Interactive Computer Support for Writing. *Cognition and Instruction* 22(3), pp. 333–362.

Whitelock, D. (2010). Activating Assessment for Learning: Are We on the Way with Web 2.0? In *Web 2.0-Based E-Learning: Applying Social Informatics for Tertiary Teaching*, eds. Mark J.W. Lee and Catherine McLoughlin. Hershey, PA: IGI Global pp. 319–342.

APPENDIX

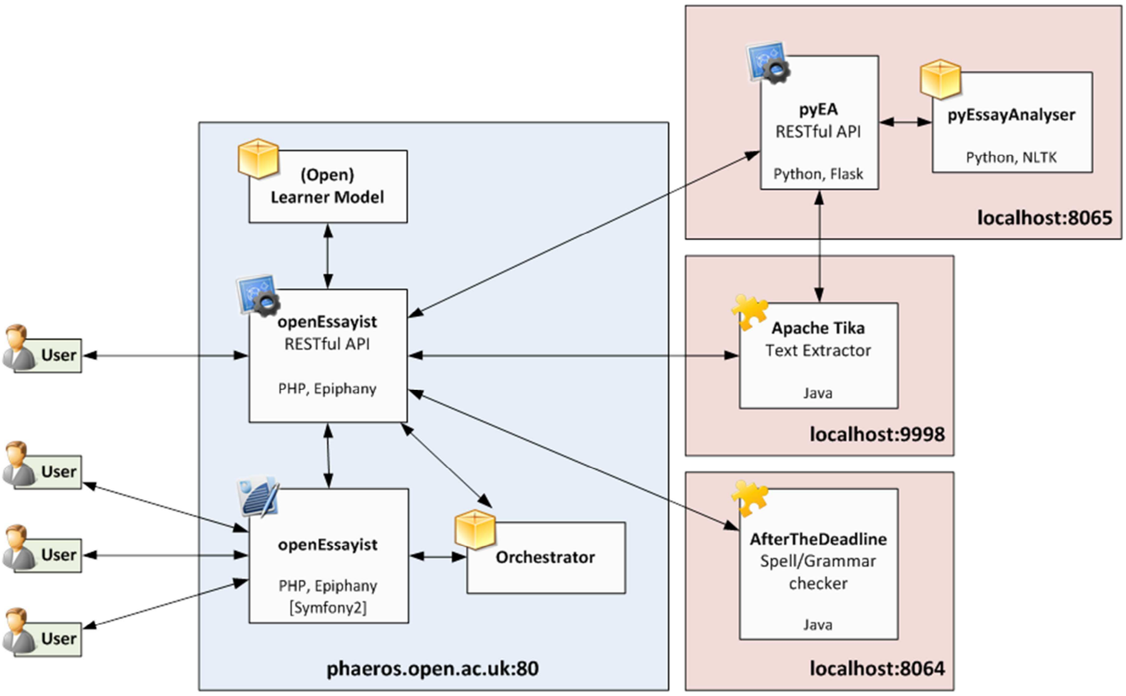
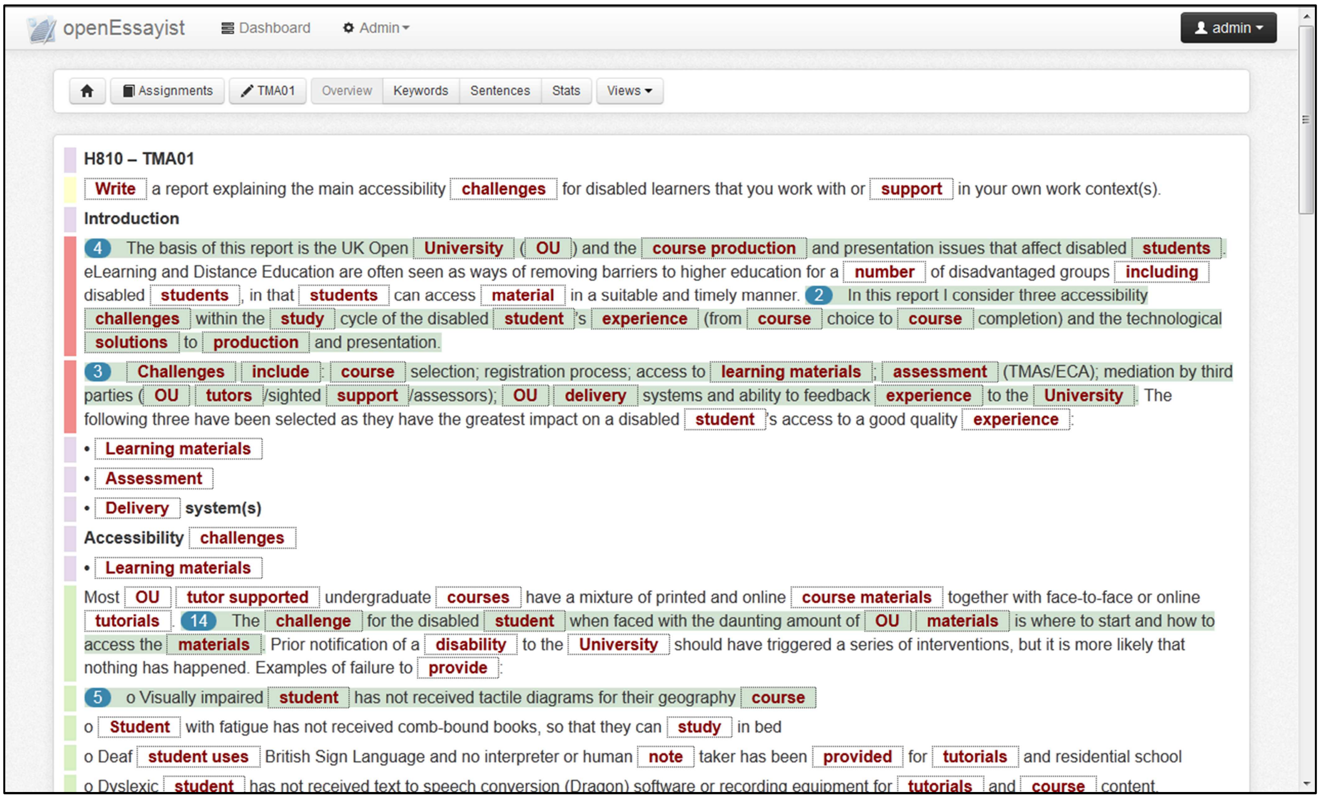


Figure 1. Architecture of OpenEssayist



H810 – TMA01

Write a report explaining the main accessibility **challenges** for disabled learners that you work with or **support** in your own work context(s).

Introduction

4 The basis of this report is the UK Open **University** (**OU**) and the **course production** and presentation issues that affect disabled **students** . eLearning and Distance Education are often seen as ways of removing barriers to higher education for a **number** of disadvantaged groups **including** disabled **students** , in that **students** can access **material** in a suitable and timely manner. 2 In this report I consider three accessibility **challenges** within the **study** cycle of the disabled **student** 's **experience** (from **course** choice to **course** completion) and the technological **solutions** to **production** and presentation.

3 **Challenges** **include** : **course** selection; registration process; access to **learning materials** ; **assessment** (TMAs/ECA); mediation by third parties (**OU** **tutors** /sighted **support** /assessors); **OU** **delivery** systems and ability to feedback **experience** to the **University** . The following three have been selected as they have the greatest impact on a disabled **student** 's access to a good quality **experience** :

- **Learning materials**
- **Assessment**
- **Delivery** system(s)

Accessibility **challenges**

- **Learning materials**

Most **OU** **tutor supported** undergraduate **courses** have a mixture of printed and online **course materials** together with face-to-face or online **tutorials** . 14 The **challenge** for the disabled **student** when faced with the daunting amount of **OU** **materials** is where to start and how to access the **materials** . Prior notification of a **disability** to the **University** should have triggered a series of interventions, but it is more likely that nothing has happened. Examples of failure to **provide** :

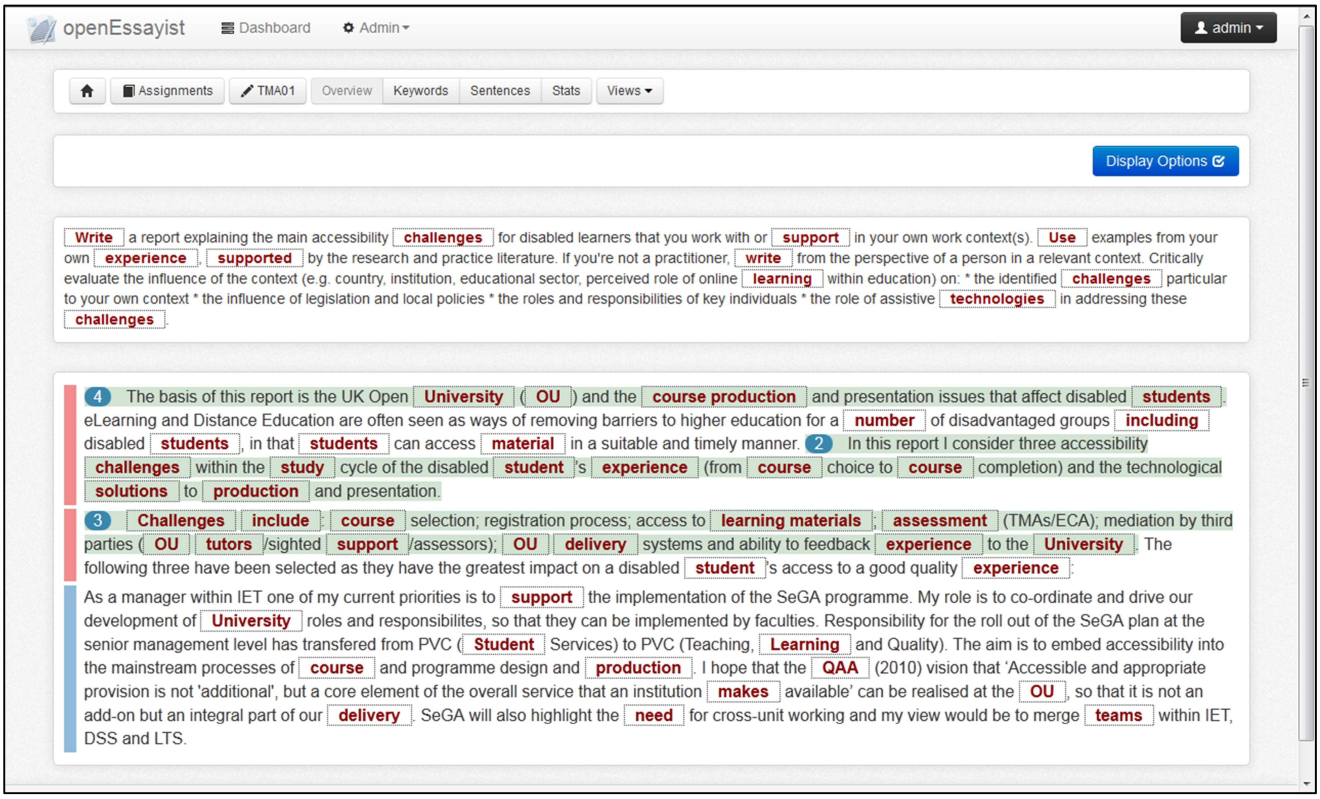
5 o Visually impaired **student** has not received tactile diagrams for their geography **course**

o **Student** with fatigue has not received comb-bound books, so that they can **study** in bed

o Deaf **student uses** British Sign Language and no interpreter or human **note** taker has been **provided** for **tutorials** and residential school

o Dyslexic **student** has not received text to speech conversion (Dragon) software or recording equipment for **tutorials** and **course** content.

Figure 2. Key words, phrases and sentences visualized in the essay context. Sentences in light-grey (green) background are key sentences as extracted by the EssayAnalyser (the number indicates its key-ness ranking). Key words and key phrases are indicated in bold (red) and boxed.



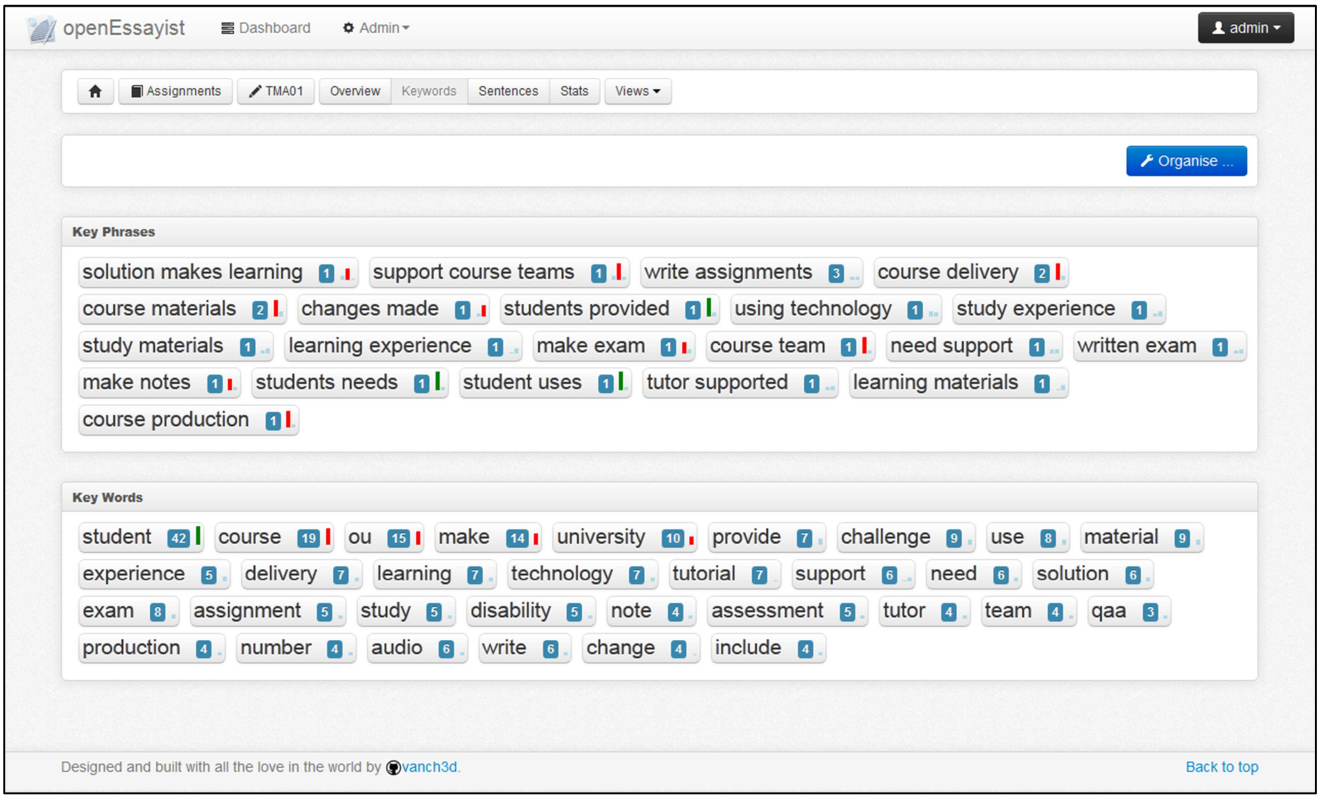
Write a report explaining the main accessibility challenges for disabled learners that you work with or support in your own work context(s). Use examples from your own experience, supported by the research and practice literature. If you're not a practitioner, write from the perspective of a person in a relevant context. Critically evaluate the influence of the context (e.g. country, institution, educational sector, perceived role of online learning within education) on: * the identified challenges particular to your own context * the influence of legislation and local policies * the roles and responsibilities of key individuals * the role of assistive technologies in addressing these challenges.

4 The basis of this report is the UK Open University (OU) and the course production and presentation issues that affect disabled students. eLearning and Distance Education are often seen as ways of removing barriers to higher education for a number of disadvantaged groups including disabled students, in that students can access material in a suitable and timely manner. 2 In this report I consider three accessibility challenges within the study cycle of the disabled student's experience (from course choice to course completion) and the technological solutions to production and presentation.

3 Challenges include: course selection; registration process; access to learning materials; assessment (TMAs/ECA); mediation by third parties (OU tutors/sighted support/assessors); OU delivery systems and ability to feedback experience to the University. The following three have been selected as they have the greatest impact on a disabled student's access to a good quality experience.

As a manager within IET one of my current priorities is to support the implementation of the SeGA programme. My role is to co-ordinate and drive our development of University roles and responsibilites, so that they can be implemented by faculties. Responsibility for the roll out of the SeGA plan at the senior management level has transfered from PVC (Student Services) to PVC (Teaching, Learning and Quality). The aim is to embed accessibility into the mainstream processes of course and programme design and production. I hope that the QAA (2010) vision that 'Accessible and appropriate provision is not 'additional', but a core element of the overall service that an institution makes available' can be realised at the OU, so that it is not an add-on but an integral part of our delivery. SeGA will also highlight the need for cross-unit working and my view would be to merge teams within IET, DSS and LTS.

Figure 3. The structural elements of the essay can be used jointly with the key word extraction to highlight relevant information within specific parts of the essay, here in both introduction and conclusion (and the assignment question).



Key Phrases

solution makes learning 1 | support course teams 1 | write assignments 3 | course delivery 2
course materials 2 | changes made 1 | students provided 1 | using technology 1 | study experience 1
study materials 1 | learning experience 1 | make exam 1 | course team 1 | need support 1 | written exam 1
make notes 1 | students needs 1 | student uses 1 | tutor supported 1 | learning materials 1
course production 1

Key Words

student 42 | course 19 | ou 15 | make 14 | university 10 | provide 7 | challenge 9 | use 8 | material 9
experience 5 | delivery 7 | learning 7 | technology 7 | tutorial 7 | support 6 | need 6 | solution 6
exam 8 | assignment 5 | study 5 | disability 5 | note 4 | assessment 5 | tutor 4 | team 4 | qaa 3
production 4 | number 4 | audio 6 | write 6 | change 4 | include 4

Designed and built with all the love in the world by vanch3d.
Back to top

Figure 4. Key words and phrases as separate lists.

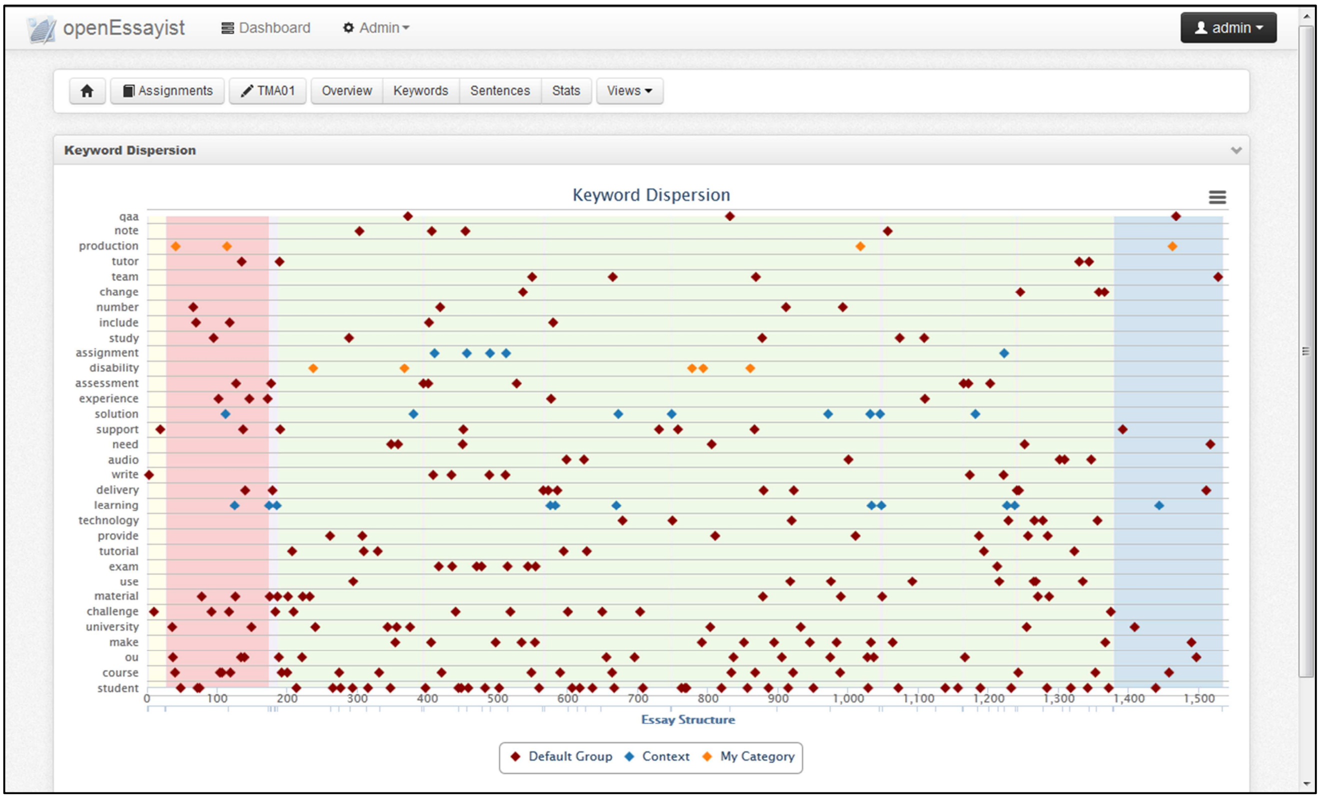
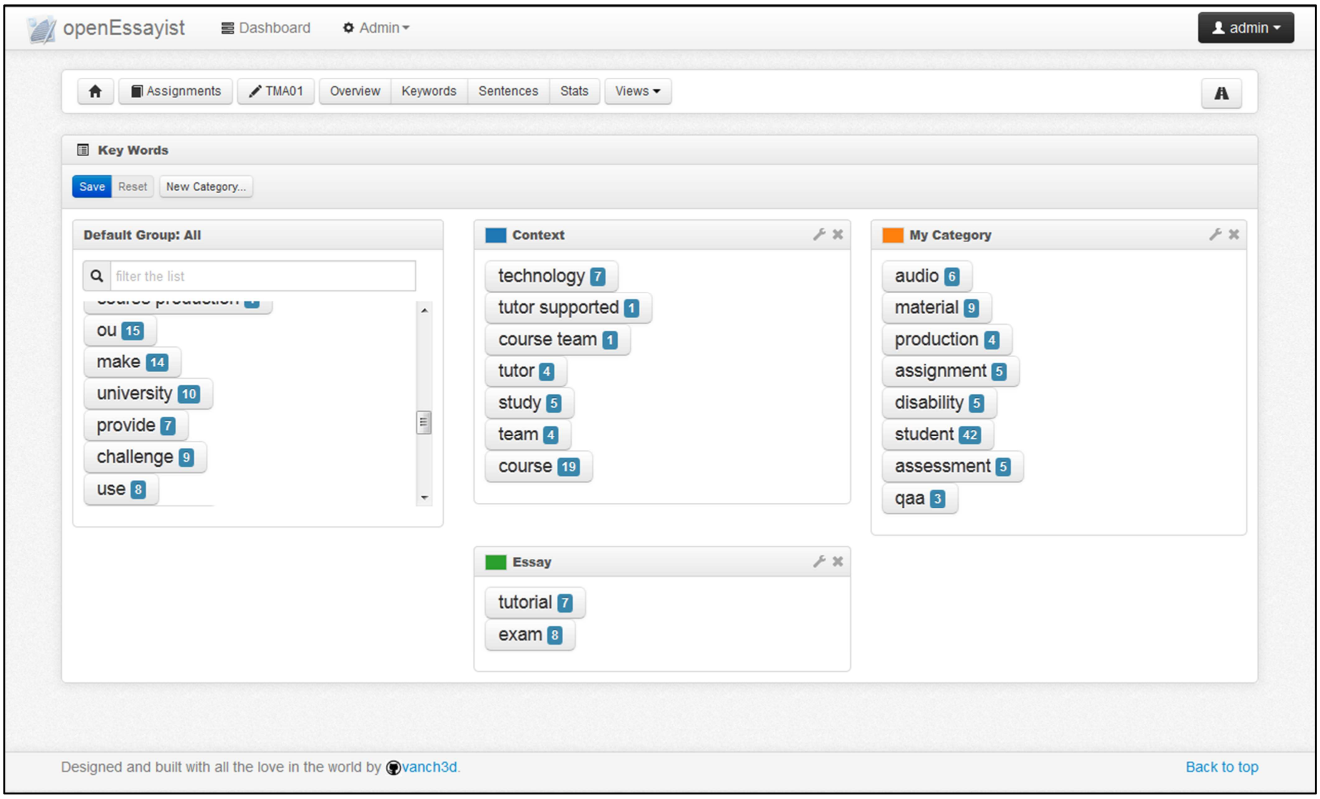


Figure 5. Dispersion of key words across the essay.



Key Words

Default Group: All

- ou 15
- make 14
- university 10
- provide 7
- challenge 9
- use 8

Context

- technology 7
- tutor supported 1
- course team 1
- tutor 4
- study 5
- team 4
- course 19

My Category

- audio 6
- material 9
- production 4
- assignment 5
- disability 5
- student 42
- assessment 5
- qaa 3

Essay

- tutorial 7
- exam 8

Figure 6. Key words extracted by the systems are re-organized by the students, using their own categories

A User Study on the Automated Assessment of Reviews

Lakshmi Ramachandran and Edward F. Gehringer
North Carolina State University
{lramach,efg}@ncsu.edu

ABSTRACT

Reviews are text-based feedback provided by a reviewer to the author of a submission. Reviews play a crucial role in providing feedback to people who make assessment decisions (e.g. deciding a student's grade, purchase decision of a product). It is therefore important to ensure that reviews are of a good quality. In our work we focus on the study of academic reviews. A review is considered to be of a good quality if it can help the author identify mistakes in their work, and help them learn possible ways of fixing them. *Metareviewing* is the process of evaluating reviews. An automated metareviewing process could provide quick and reliable feedback to reviewers on their assessment of authors' submissions. Timely feedback on reviews could help reviewers correct their assessments and provide more useful and effective feedback to authors. In this paper we investigate the usefulness of metrics such as *review relevance, content type, tone, quantity* and *plagiarism* in determining the quality of reviews. We conducted a study on 24 participants, who used the automated assessment feature on Expertiza, a collaborative peer-reviewing system. The aim of the study is to identify reviewers' perception of the usefulness of the automated assessment feature and its different metrics. Results suggest that participants find relevance to be the most important and quantity to be the least important in determining a review's quality. Participants also found the system's feedback from metrics such as content type and plagiarism to be most useful and informative.

Keywords

review quality assessment, metareview metrics, user experience survey

1. INTRODUCTION

In recent years there has been a considerable amount of research directed towards developing educational systems that foster collaborative learning. Collaborative learning systems provide an environment for students to interact with other students, exchange ideas, provide feedback and use the feedback to improve their own work. Systems such as SWoRD [1] and Expertiza [3] are web-based collaborative peer-review systems, which promote team work by allowing students to build shared knowledge with an exchange of ideas. These systems also provide an environment for students to give feedback to peers on their work.

The process of providing feedback to peers on their work may help students learn more about the subject, and develop their critical thinking. Rada et al. found that students who evaluated their peers' work were more likely to improve the quality of their own work than those students who did not provide peer reviews [4]. The peer review process may also help students learn to be more responsible.

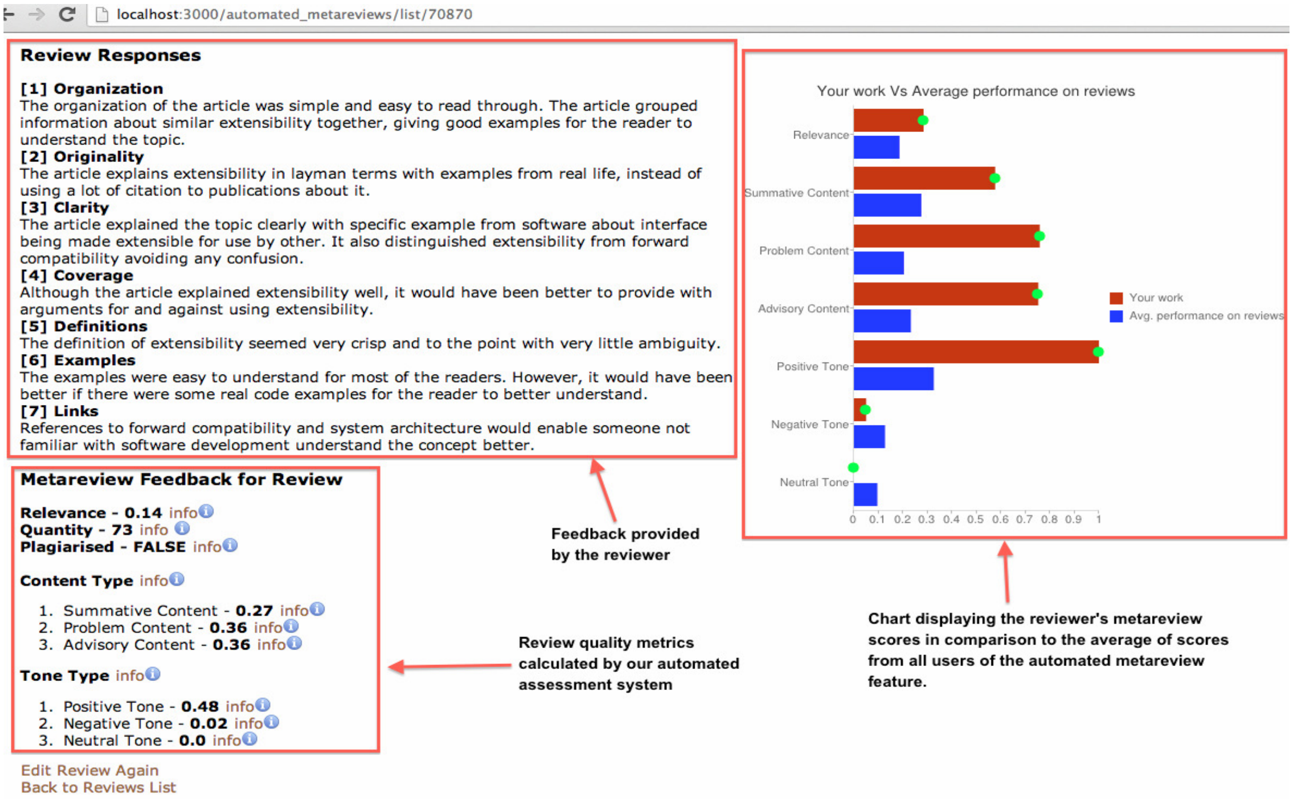
The classroom peer review process is very much similar to the process of reviewing scientific articles for journals. Scientific reviewers tend to have prior reviewing experience and a considerable knowledge in the area of the author's submission (the text under review). Students on the other hand are less likely to have had any prior reviewing experience. They have to be guided to provide good quality reviews that may be useful to their peers.

Metareviewing can be defined as the process of reviewing reviews, i.e., the process of identifying the quality of reviews. Metareviewing is a manual process [2, 5, 6] and just as with any process that is manual; metareviewing too is (a) slow, (b) prone to errors and (c) likely to be inconsistent - the set of problems, which makes automated metareviewing necessary. An automated metareview process ensures consistent, bias-free reviews to all reviewers. This also ensures provision of immediate feedback to reviewers, which is likely to motivate them to improve their work and provide more useful feedback to the authors.

In this work we propose the use of a system that automatically evaluates student review responses. We use a specific set of metrics such as *review's relevance* to the work under review (or the submission), the *type of content* a review contains, *tone* of the review, *quantity* of feedback provided and presence of *plagiarism*, to carry out metareviewing. We have integrated the automated metareview feature (with the listed set of metrics) into Expertiza [3]. Expertiza is a collaborative web-based learning application. A screenshot of the metareview output from the system is shown in Figure1. We have conducted an exploratory analysis to study the importance of the review quality metrics and usefulness of the system's outputs, as judged by users of the system.

2. RELATED WORK

One of the earlier approaches to manually assessing the quality of peer reviews involved the creation and use of a Review Quality Instrument (RQI) [9]. Van Rooyen et al. use the RQI to check whether a reviewer discusses the following - (1) importance of the



Review Responses

[1] Organization
The organization of the article was simple and easy to read through. The article grouped information about similar extensibility together, giving good examples for the reader to understand the topic.
[2] Originality
The article explains extensibility in layman terms with examples from real life, instead of using a lot of citation to publications about it.
[3] Clarity
The article explained the topic clearly with specific example from software about interface being made extensible for use by other. It also distinguished extensibility from forward compatibility avoiding any confusion.
[4] Coverage
Although the article explained extensibility well, it would have been better to provide with arguments for and against using extensibility.
[5] Definitions
The definition of extensibility seemed very crisp and to the point with very little ambiguity.
[6] Examples
The examples were easy to understand for most of the readers. However, it would have been better if there were some real code examples for the reader to better understand.
[7] Links
References to forward compatibility and system architecture would enable someone not familiar with software development understand the concept better.

Metareview Feedback for Review

Relevance - 0.14 info
Quantity - 73 info
Plagiarised - FALSE info

Content Type info

1. Summative Content - **0.27** info
2. Problem Content - **0.36** info
3. Advisory Content - **0.36** info

Tone Type info

1. Positive Tone - **0.48** info
2. Negative Tone - **0.02** info
3. Neutral Tone - **0.0** info

Edit Review Again
Back to Reviews List

Feedback provided by the reviewer

Review quality metrics calculated by our automated assessment system

Chart displaying the reviewer's metareview scores in comparison to the average of scores from all users of the automated metareview feature.

Figure 1: Output from the automated metareview feature on Expertiza [3]. We provide a comparison of the participant reviewer's scores with other reviewers' metareview scores (in a chart) to help reviewers gauge how well they are doing on a certain metric.

research question, (2) originality, (3) strengths and weaknesses, (4) presentation and interpretation of results. In addition, the RQI also checks if a review was constructive, and if the reviewer had substantiated his/her claims. We incorporate some of these metrics in our approach, e.g. detecting constructiveness in reviews (based on its content), checking whether reviewers substantiated their claims (by identifying relevance to the author's submission), to automatically assess review quality.

Nelson and Schunn studied feedback features that help authors understand and use reviews [10]. They found that features such as problem localization and solution suggestion helped authors understand feedback. These are some of the types of content we look for during review content identification.

Kuhne et al. use authors' ratings of reviews to identify the quality of peer reviews [5]. They found that authors are contented with reviewers who appear to have made an effort to understand their work. This finding is useful to our automatic review quality assessment system, which assesses reviews based on the usefulness of its content. Our system also detects the relevance of reviews, which may be indicative of the effort made by a reviewer to understand the author's work and provide specific feedback.

Xiong et al. look for problems identified by reviewers in the author's work in peer reviews from the SWoRD system [11]. Xiong et al. use a bag-of-words, exact match approach to detect problem localization features. They use a shallow semantic match approach, which uses counts of nouns, verbs etc. in the text as features. Their approach does not incorporate relevance identification nor does it identify content type. Cho uses machine classification techniques such as naïve Bayes, support vector machines (SVM) and decision trees to classify review comments [12]. Cho manually breaks down every peer comment into idea units, which are then coded as a praise, criticism, problem detection, solution suggestion, summary or off-task comment.

Some other approaches used to study the usefulness of reviews are those by Turney [15], Dalvi [16] and Titov [17]. Peter D. Turney uses semantic orientation (positive or negative) to determine whether a review can be classified as recommended or not recommended. Turney's approach to differentiate positive from negative reviews involves identifying similarity between phrases containing adverbs and adjectives to terms “excellent" and “poor". Turney uses semantic orientation to recommend products or movies. We also use semantic orientation (referred to as tone) to identify the degree of sensitivity with which reviewers convey their criticisms.

Lim et al. identify reviewers who target e-commerce products and applications, and generate spam reviews [18]. The problem of spamming may be analogous to the problem of copy-pasting text in order to game the automated assessment system into giving reviewers high scores on their reviews. Therefore, we use a metric to detect plagiarized reviews.

There exist research works that discuss metrics that are important in review quality identification, and some that apply shallow approaches to determine quality. However, there is no work that takes factors such as relevance, content type, tone, quantity and plagiarism into consideration while determining review quality. Our sys-

Table 1: Some examples of reviews.

S No.	Review
1	“The example needs work."
2	“Yes, the organization is poor."

tem is an amalgamation of existing research in the said areas. In the next section, we provide an overview of the different review quality metrics.

3. AUTOMATED REVIEW ASSESSMENT

In order to assess quality, reviews have to be represented using metrics that capture their most important features. In general a good review contains: (1) coherent and well-formed sentences, which can be easily comprehended by the author, and (2) sufficient amount of feedback. In this section we discuss the metrics we use to assess reviews.

3.1 Review relevance

Reviewers may provide vague, unjustified comments. Comments in Table 1 are generic, and do not refer to a specific object in the text under review. For instance, what type of “work" does the “example" need or, what is poor about the “organization"? These reviews are ambiguous, and need to be supported with more information. Also, how do we know if the review has been written for the right submission, for instance any article may contain an example. Review relevance helps identify if a review is talking about the right submission.

We identify relevance in terms of the semantic and syntactic similarities between two texts. We use a word order graph, whose vertices, edges and double edges help determine structure-based match across texts. We use WordNet to determine semantic relatedness. Our approach has been described in Ramachandran and Gehringer [19].

3.2 Review content

A review is expected to provide an assessment of the kind of work that was done - praising the submission’s positive points, identifying problems, if any, and offering suggestions on ways of improving the submission. Review examples in Table 1 do not provide any details. Reviews must identify problems in the author’s work, and provide suggestions for improvement in order to be useful to authors, thus helping them understand where their work is lacking or how it can be improved. Content of a review identifies the type of feedback provided by the reviewer. We look for the following types of content in a review:

- **Summative** - Summative reviews contain either a positive or a neutral assessment of the author’s submission. *Example:* “I guess a good study has been done on the tools as the content looks very good in terms of understanding and also originality. Posting reads well and appears to be largely original with appropriate citation of other sources."
- **Problem detection** - Reviews in this category are critical of the author’s submission and point out problems in the submission. *Example:* “There are few references used and there are sections of text quoted that appear to come from a multitude of web sites." However, problem detection reviews only find problematic instances in the author’s work, and do not offer any suggestions to improve the work.
- **Advisory** - Reviews that offer the author suggestions on ways of improving their work fall into this category. *Example:* “Although the article makes use of inline citations which is a plus, there are only a few references. Additional references could help support the content and potentially provide the examples needed." Advisory reviews display an understanding of the author’s work, since the reviewer has taken the effort to provide the author with constructive feedback.

Different types of review content have different degrees of usefulness. For instance summative reviews provide only summaries of the author’s work and are less useful to the author, whereas reviews that identify problems in the author’s work or provide suggestions can be used by authors to improve their work, and are hence considered more important. We use a cohesion-based pattern identification technique to capture the meaning of a class of reviews.

3.3 Review tone

Tone refers to the semantic orientation of a text. Semantic orientation depends on the reviewer’s choice of words and the presentation of a review. Tone of a review is important because while providing negative criticism reviewers might unknowingly use words that may offend the authors. Therefore we use tone information to help guide reviewers. A review can have one of three types of tones - positive, negative or neutral. We look for positively or negatively oriented words to identify the tone of a review [15]. We use positive and negative indicators from an opinion lexicon provided by Liu et al [20] to determine the semantic orientation of a review. Semantic orientation or tone of the text can be classified as follows:

- **Positive** - A review is said to have a positive tone if it predominantly contains positive feedback, i.e., it uses words or phrases that have a positive semantic orientation. *Example:* “The page is very well-organized, and the information is complete and accurate." Adjectives such as “well-organized", “complete" and “accurate" are good indicators of a positive semantic orientation.
- **Negative** - This category contains reviews that predominantly contain words or phrases that have a negative semantic orientation. Reviews that provide negative criticism to the author’s work fall under this category, since while providing negative remarks reviewers tend to use language that is likely to offend the authors. Such reviews could be morphed or written in a way that is less offensive to the author of a submission. *Example:* “The examples are not easy to understand and have been copied from other sources. Although the topic is Design Patterns in Ruby, no examples in Ruby have been provided for Singleton and Adapter Pattern."

  The given example contains negatively oriented words or phrases such as “not easy to understand" ,“copied", “no examples". Review segment “..have been copied from other sources..." implies that the author has plagiarized, and could be construed by the author as a rude accusation. One of the ways in which this review could be re-phrased to convey the message, so as to get the author to acknowledge the mistake and make amends, is as follows - “The topic on Design Patterns in Ruby could be better understood with more examples, especially for the Singleton and Adapter patterns. Please try to provide original examples from your experience or from what was discussed in class."

- **Neutral** - Reviews that do not contain either positively or negatively oriented words or phrases, or contain a mixture of both are classified into this category. *Example:* “The organization looks good overall. But lots of IDEs are mentioned in the first part and only a few of them are compared with each other. I did not understand the reason for that." This review contains both positively and negatively oriented segments, i.e., “The organization looks good overall" is positively oriented, while “I did not understand the reason for that." is negatively oriented. The positive and negatively oriented words when taken together give this review a neutral orientation.

In case of both content and tone, a single review may belong to multiple categories. For instance consider the review, “Examples provided are good; a few other block structured languages could have been talked about with some examples as that would have been pretty useful to give a broader pool of languages that are block structured.” While classifying for content, we see that the first part of the review, “Examples provided are good” praises the submission, while the remaining part of the review provides advice to the author. Our content identification technique identifies the amount of each type of content or tone (on a scale of 0 - 1) a review contains. Similarly in the case of tone, we identify the degree of positive, negative or neutral orientation of each review.

3.4 Review quantity

Text quantity is important in determining review quality since a good review provides the author with sufficient feedback. We plan on using this metric to indicate to the reviewer the amount of feedback they have provided in comparison to the average review quantity (from other reviewers of the system), thus motivating reviewers to provide more feedback to the authors. We identify quantity by taking a count of all the unique tokens in a piece of review. For instance, consider the following review, “The article clearly describes its intentions. I felt that section 3 could have been elaborated a little more." The number of unique tokens in this review is 15 (excluding articles and pronouns).

3.5 Plagiarism

Reviewers tend to refer to content in the author’s submission in their reviews. Content taken from the author’s submission or from some external source (Internet) should be placed within quotes in the review. If reviewers copy text from the author’s submission and fail to place it within quotes (knowingly or unknowingly) it is considered as *plagiarism*.

Each of the review quality metrics listed is determined independently, and are integrated into a complete review quality assessment system. Reviewers are given feedback on each of these listed metrics, so that they get a complete picture of the completeness and quality of their review.

4. USER EXPERIENCE STUDY

We decided to study the experience of using an automated metareview system, since different types of reviewers - students, teaching assistants and faculty may use this feature. We study the extent to which users of an automated quality assessment system would perceive it to be important, and the output of the system to be useful. The study is important because it helps us understand whether reviewers learn and benefit from such an automated metareview system. This study also helps us learn what aspects of the feature can be improved, by identifying what the surveyed reviewers liked or disliked about the feature. A positive experience from using this feature may mean that reviewers would be more inclined to use it to improve their reviews.

According to Kuniavsky [21], user experience is “the totality of end-users’ perceptions as they interact with a product or service. These perceptions include effectiveness (how good is the result?), efficiency (how fast or cheap is it?), emotional satisfaction (how good does it feel?), and the quality of the relationship with the entity that created the product or service (what expectations does it create for subsequent interactions?).” There exist several other definitions for the term *user experience* (abbreviated as UX) [22]. UXMatters[1] defines user experience as that which “Encompasses all aspects of a digital product that users experience directly - and perceive, learn, and use - including its form, behavior, and content.” They also state that “Learnability, usability, usefulness, and aesthetic appeal are key factors in users’ experience of a product.” Therefore, apart from a study of factors such as user’s perceptions, feelings or responses to a system, a user experience survey should also involve a study of the learning gained from a system and the usefulness of a system.

The aim of this study is to identify the degree of importance participants attach to each of the metareview metrics–review relevance, content, tone, quantity and plagiarism. This study will help us identify how effective the system is at helping reviewers learn about characteristics of their reviews.

5. EXPERIMENTS

To study the usefulness of our review quality assessment system we investigate the following broad research questions:

RQ1: *Do automated metareviews provide useful feedback?*
RQ2: *Which of the review quality metrics are more or less important than the others?*
RQ3: *Which of the review quality metrics’ output did the reviewers find more or less useful when compared to the others?*

5.1 Participants

In order to identify how useful users of the automated metareview feature find it to be, we recruited 24 participants to (1) use the feature on Expertiza and (2) provide us with information on their experience by filling out a survey. Participants were recruited with an email message, which explained to them the purpose of the study. The set of participants included 15 doctoral students, 3 masters’ students and 1 undergraduate student, all of whom were from the computer science department at North Carolina State University, and 5 research scientists from academia and industry.

5.2 Data collection

Our data collection process involved two steps. In the first step, participants were asked to use the automated metareview feature on Expertiza. They use the system to write a review for an article. For our study, we chose a wiki article on *Software Extensibility*[2]. We chose this article since we were recruiting subjects from the field of computer science, and Software Extensibility is a topic most computer science students or researchers are familiar with. A detailed

[1] UXMatters - User experience definition - http://www.uxmatters.com/glossary/

[2] Software Extensibility - https://en.wikipedia.org/wiki/Extensibility

Table 2: Detailed set of instructions to help complete the survey

1. Use username/password to log into Expertiza.
2. Click on assignment “User Study”
3. Click on “Others’ Work” (Since you will be reviewing someone else’s work.)
4. Click on “Begin” to start the review.
5. Click the url under the “Hyperlinks” section. Read the article on Software Extensibility. Please keep in mind that you are reviewing this article.
6. Answer questions on the review rubric describing the quality of the article you read. After answering all the review questions, click on the “Save Review” button.
7. Wait for a few minutes for the system to generate the automated feedback.
8. Fill out the **user-experience questionnaire**.

set of instructions was provided to each of the participants to help them complete the study (Table 2).

A review rubric is provided to the participants to help them write the review. The rubric contains questions on the organization, originality, clarity and coverage of the article under review. The rubric also evokes information on quality of the definitions, examples and links found in the article.

When participants submit their reviews, they are presented with automated feedback from our system. This feedback gives them information on different aspects of their review such as (1) content type, (2) relevance of the review to the article, (3) tone, (4) quantity of text and (5) presence of plagiarism. A screenshot of the output is available in Figure1. The participant reviewer reads and understands the metareview feedback.

In the second step of data collection, the participant reviewer is asked to fill out a user experience questionnaire (Step 8 in Table 2). The user experience questionnaire is a big part of this study, and has been explained in detail in Section 6.

6. USER EXPERIENCE QUESTIONNAIRE

The user experience questionnaire consists of four sections - *participant background, importance of reviews, importance of metrics, usefulness of system’s output*. The questions we use in our user experience survey are discussed in the following sections. Answers to each of these questions are given on a scale of 1 (lowest) to 5 (highest).

6.1 Participant background

In the *background* section, participants were questioned about their experience in writing reviews, and in their experience with using peer-review systems such as Expertiza. The exact questions were:

Q1: *Do you have prior reviewing experience?*
Q2: *Do you have prior experience using the Expertiza system?*
Q3: *Have you used a peer-review system before?*
Q4: *Are you a(n): Undergraduate, Masters or PhD student, or Other?*

6.2 Importance of reviews and metareviews

In the *importance* section, we questioned participants on the importance of reviews and metareviews to a system.

Q5: *How important do you think reviews are in a decision-making process?*
Q6: *How important do you think metareviews (review of a review) are in a decision-making process?*

Answers are given on a 5-point scale - *unimportant, somewhat important, neutral, important* and *extremely important*. This section also includes an open question to gather textual feedback from participants. All these questions are optional, i.e., the participant may choose not to respond to any of them.

We also gauge whether participants would be motivated to use reviews to improve the quality of their submission (as an author), and metareviews to improve the quality of their reviews (as a reviewer). We therefore included the following questions in the questionnaire:
Q7: *Would better reviews inspire you to use the feedback in your revisions?*
Q8: *Would automated metareviews motivate you to update your reviews?*
Q9: *Do the automated metareviews provide useful feedback?*

6.3 Importance of metareview metrics

In the *importance of metrics* section we identify how important participants think the different metareview metrics are in gauging the quality of a review.

Q10: *How important do you think each of the review quality metrics is in learning about the quality of your review? 1. Review relevance, 2. Review content 3. Tone 4. Quantity 5. Plagiarism*

The answers are given on a 5-point scale. This question helps us identify the metrics to which users of the system attach most importance, or to which ones they attach the least importance. This section also allows participants to provide any additional comments, to learn about the participants’ opinions of the different metrics, or any other related information.

6.4 Usefulness of system’s metareview output

This section helps us study the usefulness of the system’s outputs. These questions gauge whether reviewers learned something about their review’s quality from the automated feedback.

Q11: *How useful do you think the output from each of the review quality metrics is (from what you saw on Expertiza)? 1. Relevance, 2. Review content 3. Tone 4. Quantity 5. Plagiarism*

Answers are given on a 5-point scale and range from *not useful, somewhat useful, neutral, useful* or extremely useful. The ratings indicate usefulness of the chosen design for the system’ output. These questions help us learn whether participants are able to successfully comprehend the meaning of the system’s output. This information coupled with the information from the previous question on *importance of metrics* would help us identify the set of metrics that need improving. This section also includes an open question to gather any other comments participants may have on the system’s output.

6.5 Other metrics

We included an open question on the survey to learn about any other review quality metrics, which participants think would be useful in an automated metareview system.

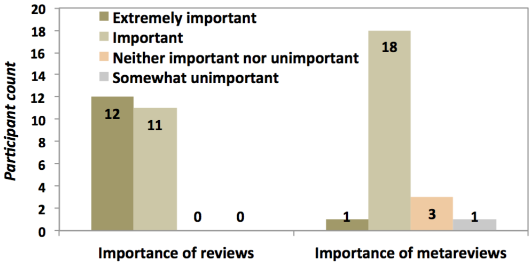


Figure 2: Participants' rating of importance of reviews and metareviews.

Q12: *What other information do you think might help you improve your review quality? Are there any specific review features you would like to get feedback on? e.g. language of the review, grammar, vocabulary, or nothing else*

The next section discusses our analyses on the collected data.

7. ANALYSIS OF DATA

In this section we discuss some of the findings from our data. Out of the 24 participants, 19 had prior reviewing experience. Only 7 of the participants had prior experience with the Expertiza system.

7.1 Importance of reviews and metareviews

All of the participants agreed that reviews play an important role in the decision-making process (Figure 2). A majority of the participants also agreed on the importance of metareviews (review of reviews). One participant did not respond to these questions.

We asked participants whether good quality reviews would motivate them to fix their submission. All participants agreed (7 agreed strongly) that they would incorporate suggestions from the feedback in their work (Figure 3). We asked participants whether automated feedback on their reviews would inspire them to improve their reviews. Out of the 24 participants 13 agreed that they would use the automated feedback. However 8 participants displayed doubt in the use of automated metareview feedback by answering *neither agree nor disagree*. A small number said that they would not be inclined to use the automated metareview feedback to improve their reviews.

Thus we see that as authors, participants agree that good quality feedback would motivate them to fix their work, but as reviewers they may not be inclined to use metareview feedback to update their reviews (and help other authors improve their work). The concept of automated assessment of reviews is new, and a lack of understanding of the purpose of these metrics could be one of the reasons why reviewers felt that automated metareviews may not motivate them to fix their reviews.

7.2 Importance of the review quality metrics

We analyze how participants judge each of the automated metrics' importance. The results are displayed in Figure 4. The metric, which participants rated as the most important is *relevance*. Out of the 24 participants 23 agree that relevance is important in assessing the quality of a review (3 thought it was extremely important). The next most important metric was found to be *review content*, with 21 of the participants agreeing on its importance (3 thought it was extremely important).

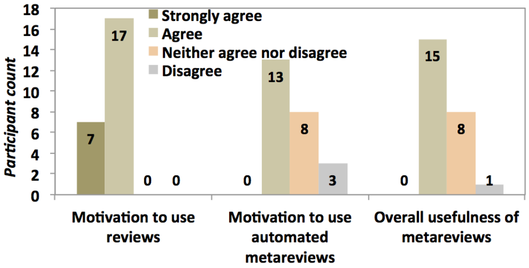


Figure 3: Participants' rating of motivation to use reviews and metareviews to improve the quality of their submission or review respectively. The chart also contains participants' estimation of usefulness of the automated metareview feature's output.

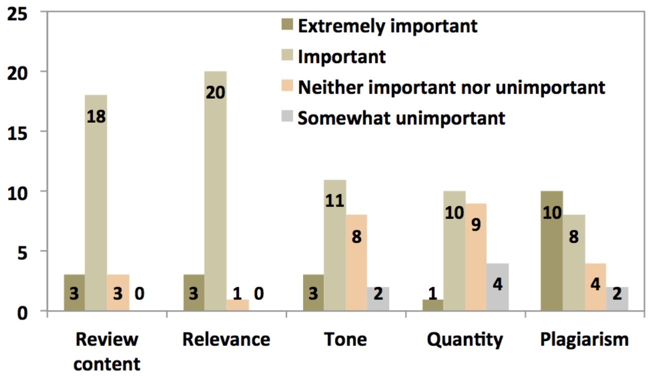


Figure 4: Participants' rating of the importance of each review quality metric.

Participants found quantity to be the least important metric, with 9 of them expressing doubts on its usefulness (neither important nor unimportant) and 4 of them describing it as somewhat unimportant. Wilcoxon rank-sum test is used to determine if two metrics' ratings have identical distributions (null hypothesis) [23]. We use this test to compare metric quantity with metrics relevance and content (which have been identified as the most important metrics) at 0.05 significance level. The p value for the test on metrics quantity and relevance is 0.0003, and for metrics quantity and content is 0.002. Since these p values are < 0.05, we conclude that quantity's ratings are significantly different from those of the most important metrics - relevance and content.

Quantity contains the number of unique tokens in a review text, and is meant to motivate reviewers to write more feedback. Quantity may be obvious to a reviewer, since they are aware of the amount of feedback they have provided. Hence quantity may turn out to be the least effective, when compared with the other metrics, in conveying any new information to the reviewer. This could be why quantity is ranked as the least important quality metric.

7.3 Usefulness of system output

We questioned participants on the usefulness of the system's metareview output, to study how informative or understandable they find

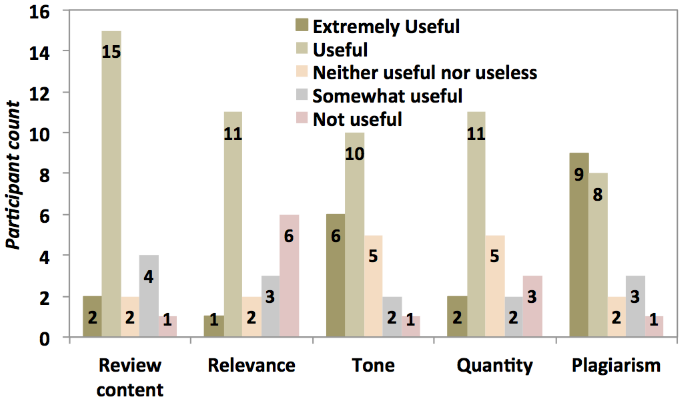


Figure 5: Participants' rating of the usefulness of each review quality metric.

it. The results of studying usefulness of metrics are displayed in Figure 5. The metrics participants rated as most useful are *plagiarism* and *review content*, with 17 of participants (9 found plagiarism extremely useful, and 2 found content extremely useful) agreeing that these metrics were useful in helping them understand where their reviews are lacking.

Tone is the second most useful metric with 16 of the participants agreeing on its usefulness, despite having 8 participants judging it to be neither important nor unimportant (from previous section). Similarly in the case of quantity, 13 of the participants found the systems' output for quantity to be useful (2 of them thought it was extremely useful), although 9 of the participants said that they thought it to be neither important nor unimportant (Figure 4).

We use the Wilcoxon test (at a significance level of 0.05) to determine if there is a significant difference (increase) in the distribution of the importance and usefulness ratings of quantity. We selected pairs, whose ratings for usefulness showed an increase from their corresponding importance ratings. The ratings have a p value of $0.03 < 0.05$, which indicates that the increase in usefulness ratings is significant. Similarly, when identifying the significance of increase between the importance and usefulness ratings of tone, we get a p value of 0.09. Although this is not < 0.05, we see that the low p value may be indicative that the improvement in usefulness ratings is not a chance occurrence (i.e., it is significant). Thus we see that although participants thought initially that tone and quantity may not be important to a metareview assessment system, they found the output from the system for these two metrics to be insightful.

Despite being judged as the most important review assessment metric only 12 of the participants found the output of the relevance metric to be useful. One of the participants expressed difficulty in interpreting the meaning of the relevance score. Our metareview feedback contains only real-valued scores in the range 0 - 1, which may not have been very useful to the reviewer in understanding the degree of relevance. This could have caused the relevance's usefulness ratings to be lower when compared to the ratings of metrics such as plagiarism, which contains true/false as output.

In the future we are planning to improve the format of the output by providing textual feedback in addition to the numeric feedback. The feedback will point to specific instances of a review that need improvement. This may make it easy for reviewers to interpret the numeric score, and maybe further motivate reviewers to use the information to improve their reviews.

7.4 Other metrics

Some of the other metrics that participants exclaimed their interest in are the *grammar and syntax* of reviews. One of the participants suggested the use of *sentence structure variability* across sentences as a means of assessing a review. The participant suggested that though short phrases may succeed in communicating the idea, they may not succeed in conveying the complete thought. The presence of well-structured sentences in a review may help the author comprehend the content of a review with ease. Well-structured sentences also indicate to authors that the reviewer put in a lot of thought and effort into writing the review. Similarly in the case of another suggested metric - *word complexity*.

Another metric suggested by a participant is *text cohesion*. Reviews sometimes contain a set of sentences, which may appear to be disconnected, i.e., lack a meaningful flow from one sentence to the next. Cohesive text help make reading and understanding reviews easier.

7.5 Usefulness of the overall automated assessment feature

We surveyed participants on the usefulness of the overall automated feedback system. Out of 24 participants 15 agreed that the feedback was useful (Figure 3), and 8 neither agreed nor disagreed.

One of the participants exclaimed concern with the use of plagiarism as a metric to assess reviews. This is likely because the participant did not see the motivation for a reviewer to plagiarize while writing reviews. Students on Expertiza are evaluated (given scores) on the quality of the reviews they write. Hence they do have a motivation to copy either other good quality reviews (available online) or chunks of text from the submission and submit them as a good quality review. Plagiarism could be caught by manual metareviewers, but may be missed by an automated system. Hence we have this additional feature to ensure that reviewers do not try to game the system by copying reviews.

8. THREATS TO VALIDITY

During the evaluation we noticed that a majority of the participants did not have prior experience in using Expertiza, which could have affected their overall performance.

We also learned, from the comments section of the questionnaire, that a few of the participants did not fully understand the meaning of some of the metrics. An understanding of the purpose of the metareview metrics is essential to assessing their importance and the output's usefulness. Hence, a lack of complete understanding of the metrics may pose as a threat to our results.

No textual reviews were provided by 4 of the participants, which means that the system outputs a value of 0 for each of the metareview metrics. Participants may not be able to discern the usefulness of metrics' outputs for which they have received a score of 0. These are some of the threats to the validity of our results.

9. FUTURE DIRECTIONS

In the future we plan on doing the following: (1) improve the display of metareview output to the reviewer, (2) identify the usefulness of other metareview metrics, (3) study the degree of agreement of the automated metareview ratings with human-provided metareview ratings, and (4) study improvement in reviewing skills.

In order to improve the system's metareview output we plan to highlight snippets of the review that need to be updated. Two participants suggested the need for additional information on metrics such as problem detection and solution suggestion. We plan to provide information on specific instances (of the author's work), which the reviewer needs to read and assess to identify problems or provides suggestions. Also, providing feedback to reviewers with samples of good quality reviews may help them learn how to fix their reviews.

We plan on investigating the use of other metrics such as sentence structure, cohesion and word complexity (discussed in Section 7.4) to study a review's quality. At present our graph-based representations capture sentence structure (e.g. subject-verb-object), but we do not study cohesion across sentences in a review. A study of cohesion may involve exploring other areas of natural language processing such as anaphora resolution [24].

We plan on investigating the extent to which the output from the automated metareview system, as a whole, agrees with human-provided values. This will help us determine whether the system would do as good a job of metareviewing i.e., be as good as human metareviewers in assessing reviews.

We would also like to study if reviewers who get feedback from the system show signs of improvement, i.e., if their reviewing skill improves with time. This would indicate that reviewers learn from the system's feedback to provide more specific and more useful reviews to authors. We would also like to investigate the impact a review quality assessment system has on the overall quality of the authors' submissions.

10. CONCLUSION

Assessment of reviews is an important problem in education, as well as science and human resources, and so it is worthy of serious attention. This paper introduces a novel review quality feature, which uses metrics such as review content type, relevance, tone, quantity and plagiarism to assess reviews. This feature is integrated into Expertiza, a collaborative web-based learning application. We surveyed 24 participants on the importance of the metrics and usefulness of the review quality assessment's output. Results indicate that participants found review relevance to be most important in assessing review quality, and system output from metrics such as review content and plagiarism to be most useful in helping them learn about their reviews.

11. REFERENCES

[1] K. Cho and C. D. Schunn, "Scaffolded writing and rewriting in the discipline: A web-based reciprocal peer review system," *Computer Education*, vol. 48, pp. 409–426, April 2007.

[2] E. F. Gehringer, L. M. Ehresman, and W. P. Conger, S.G., "Reusable learning objects through peer review: The expertiza approach," in *Innovate: Journal of Online Education*, 2007.

[3] E. F. Gehringer, "Expertiza: Managing feedback in collaborative learning." in *Monitoring and Assessment in Online Collaborative Environments: Emergent Computational Technologies for E-Learning Support*, 2010, pp. 75–96.

[4] R. Rada, A. Michailidis, and W. Wang, "Collaborative hypermedia in a classroom setting," *J. Educ. Multimedia Hypermedia*, vol. 3, pp. 21–36, January 1994.

[5] C. KÃijhne, K. BÃűhm, and J. Z. Yue, "Reviewing the reviewers: A study of author perception on peer reviews in computer science." in *CollaborateCom'10*, 2010, pp. 1–8.

[6] P. Wessa and A. De Rycker, "Reviewing peer reviews: a rule-based approach," in *International Conference on E-Learning (ICEL)*, 2010, pp. 408–418.

[7] J. Burstein, D. Marcu, and K. Knight, "Finding the write stuff: Automatic identification of discourse structure in student essays," *IEEE Intelligent Systems*, vol. 18, pp. 32–39, January 2003.

[8] P. W. Foltz, S. Gilliam, and S. A. Kendall, "Supporting content-based feedback in online writing evaluation with LSA," *Interactive Learning Environments*, vol. 8, pp. 111–129, 2000.

[9] S. van Rooyen, N. Black, and F. Godlee, "Development of the review quality instrument (rqi) for assessing peer reviews of manuscripts," *Journal of Clinical Epidemiology*, vol. 52, no. 7, pp. 625 – 629, 1999.

[10] M. M. Nelson and C. D. Schunn, "The nature of feedback: How different types of peer feedback affect writing performance," in *Instructional Science*, vol. 27, 2009, pp. 375–401.

[11] W. Xiong, D. J. Litman, and C. D. Schunn, "Assessing reviewer's performance based on mining problem localization in peer-review data." in *EDM*, 2010, pp. 211–220.

[12] K. Cho, "Machine classification of peer comments in physics," in *Educational Data Mining*, 2008, pp. 192–196.

[13] R. Zhang and T. Tran, "Review recommendation with graphical model and em algorithm," in *Proceedings of the 19th international conference on World wide web*, ser. WWW '10, 2010, pp. 1219–1220.

[14] S. Moghaddam, M. Jamali, and M. Ester, "Review recommendation: personalized prediction of the quality of online reviews," in *Proceedings of the 20th ACM international conference on Information and knowledge management*, ser. CIKM '11, 2011, pp. 2249–2252.

[15] P. D. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in *In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 2002.

[16] N. Dalvi, R. Kumar, B. Pang, and A. Tomkins, "Matching reviews to objects using a language model," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, ser. EMNLP '09, 2009, pp. 609–618.

[17] I. Titov and R. McDonald, "Modeling online reviews with multi-grain topic models," in *Proceedings of the 17th international conference on World Wide Web*, ser. WWW '08, 2008, pp. 111–120.

[18] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, "Detecting product review spammers using rating behaviors," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, ser. CIKM '10, 2010, pp. 939–948.

[19] L. Ramachandran and E. F. Gehringer, "A word-order based graph representation for relevance identification [poster]," *CIKM 2012, 21st ACM Conference on Information and Knowledge Management*, October 2012.

[20] B. Liu, M. Hu, and J. Cheng, "Opinion observer: analyzing and comparing opinions on the web," in *Proceedings of the 14th International Conference on World Wide Web*, 2005, pp. 342–351.

[21] M. Kuniavsky, *Smart Things: Ubiquitous Computing User Experience Design: Ubiquitous Computing User Experience Design*. Morgan Kaufmann, 2010.

[22] E. L.-C. Law, V. Roto, M. Hassenzahl, A. P. Vermeeren, and J. Kort, "Understanding, scoping and defining user experience: a survey approach," in *Proceedings of the 27th international conference on Human factors in computing systems*. ACM, 2009, pp. 719–728.

[23] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[24] E. Tognini-Bonelli, "Corpus linguistics at work," *Computational Linguistics*, vol. 28, no. 4, pp. 583–583, 2002.

Effects of Automatically Generated Hints on Time in a Logic Tutor

First Author
University
Address
Address
Email@email.com

Second Author
University
Address
Address
Email@email.com

Third Author
University
Address
Address
Email@email.com

ABSTRACT

This work explores the effects of using automatically generated hints in Deep Thought, a propositional logic tutor. Generating hints automatically removes a large amount of development time for new tutors, and it also useful for already existing computer-aided instruction systems that lack intelligent feedback. We focus on a series of problems, after which, the control group is known to be 3.5 times more likely to cease logging onto an online tutor when compared to the group who were given hints. We found a consistent trend in which students without hints spent more time on problems when compared to students that were provided hints. Exploration of the interaction networks for these problems revealed that the control group often spent this extra time pursuing buggy-strategies that did not lead to solutions.

1. INTRODUCTION

Problem solving is an important skill across many fields, including science, technology, engineering, and math (STEM). Working open-ended problems may encourage learning in higher 'levels' of cognitive domains [2]. Intelligent tutors have been shown to be as effective as human tutors in supporting learning in many domains, in part because of their individualized, immediate feedback, enabled by expert systems that diagnose student's knowledge states [10]. However, it can be difficult to build intelligent support for students in open problem-solving environments. Intelligent tutors require content experts and pedagogical experts to work with tutor developers to identify the skills students are applying and the associated feedback to deliver [7].

In problem solving environments where students complete many diverse steps to solve a single problem, even labeling all correct and incorrect approaches is a large burden. There are many computer-based educational problem-solving environments, that have already been developed and can benefit from data-driven approaches to providing intelligent feedback. We hope to contribute toward data-driven techniques to automatically generate intelligent feedback based on previously recorded data from such environments, as well as methods to visualize and analyzes the large amounts of data present in student-log files.

Barnes and Stamper built an approach called the Hint Factory to use student data to build a graph of student problem-solving approaches that serves as a domain model for automatic hint generation [8]. Hint factory has been applied across domains [6]. Stamper et al. found that the odds of a student in the control group dropping out of the tutor were 3.5 times more likely when compared to the group provided with automatically generated hints [9]. The hints also affected problem completion rates, with the number of problems completed in L1 being significantly higher for the hint group by half of a standard deviation, when compared to the control group. Eagle and Barnes have abstracted this domain model into an Interaction Network for problem-solving data analysis. Their preliminary results show that applying graph mining techniques to Interaction Networks can help uncover useful clusters that represent diverse student approaches to solving a particular problem [5].

2. THE DEEP THOUGHT TUTOR

We perform our analysis on data from the Deep Thought propositional logic tutor [3]. Each problem provides the student with a set of premises, and a conclusion, and asks students to prove the conclusion by applying logic axioms to the premises. Deep Thought allows students to work both forward and backwards to solve logic problems [4]. Working backwards allows a student to propose ways the conclusion could be reached. For example, given the conclusion B, the student could propose that B was derived using Modus Ponens (MP) on two new, unjustified propositions: $A \to B, A$. This is like a conditional proof in that, if the student can justify $A \to B$ and A, then the proof is solved. At any time, the student can work backwards from any unjustified components, or forwards from any derived statements or the premises.

2.1 Data

We perform our experiments on the Spring and Fall 2009 Deep Thought logic tutor dataset as analyzed by Stamper, Eagle, and Barnes in 2011[9]. In this dataset, three different professors taught two semesters each of an introduction to logic course, with each professor teaching one class with hints available and one without hints in the Deep Thought tutor. In the spring semester there were 82 students in the Hint group and 37 students in the Control group; in the fall semester there were 39 students in the Hint group and 83 in the Control group. Students for which application log-data did not exist were dropped from the study; resulting in 68 and 37 students in the Hint group, and 28 and 70 students in the Control group for the first and second semesters respectively. This results in a total of 105 students in the Hint group and 98 students in the Control group. Students from the 6 sections of an introduction to logic course were assigned 13 logic proofs in the deep thought tutor. The problems are organized into three constructs: level one (L1) consisting of the first 6 problems assigned; level two (L2) consisting of the next 5 problems assigned; and level three (L3) consisting of the last two problems assigned. We refer to the group that received hints as the Hint group, and the group that did not receive hints as the Control group.

We are interested in the usage of hints from students in the hint group. Deep Thought has been modified to include John Stamper's Hint Factory [1], and provides four levels of automatically generated hints. The first level suggests the premise to be used, the second level provides more content, the third level provides the logic rule to be applied, and the fourth hint is the bottom-out hint explaining the exact procedure. We investigated two different components regarding hint usage in Deep Thought. The first is the average number of hints per level, per problem. That is, for example, the number of level two hints requested on problem 1-4. We also investigated hint coverage in the Deep Thought tutor as provided by the Hint Factory for each problem and the overall. In Deep Thought, the Hint Factory can either generate a hint, in which case all four levels of hints are generated or a hint cannot be generated in which case no hints will exist for some given step in the problem.

3. RESULTS

In order to investigate the increased rate of drop-out between the hint group and the control group. We concentrate on the first 5 problems from L1 of the Deep Thought Tutor. We focus here as, while the groups started with similar completion and attempt rates, after level 1 the groups diverge on both completion and problem attempt rates. Since investigation of the interaction networks for these problems revealed that the control group often pursue buggy-strategies, which do not result in solving the problem, we hypothesized that their would be differences in the amount of time spent in tutor between the groups.

We performed analysis on the student-tutor interaction logs. For each student we calculated the summation of their elapsed time per interaction. To control for interactions in which the student may have idled we filtered any interactions that took longer than 10 minutes. The descriptive statistics for this are located in table 1, Prob represents the problem number, H and C represent the Hint group and the Control group.

The large standard deviations are a sign that perhaps this data is not normal. Exploring the data with Q-Q plots reveals that the data is in fact, not normally distributed. This prevents us from performing between-group statistical tests, such as the student's t-test, as our data violates the assumption of normality. To normalize the data, we use a logarithmic transformation (common log) to make the data more symmetric and homoscedastic. Observation of the Q-Q plot and histogram of the transformed data reveal that we had addressed the normality concerns. The results are presented in table 2.

Table 1: Descriptive Statistics for Time (in seconds) Spent in Each Problem

	N		M		SD	
Prob	H	C	H	C	H	C
1.1	104	93	765.89	1245.24	956.41	1614.30
1.2	88	76	761.65	1114.37	911.24	1526.91
1.3	90	67	664.17	1086.09	733.95	2119.19
1.4	87	71	754.60	1266.39	1217.06	1808.53
1.5	84	67	710.62	1423.22	1192.43	2746.54

Table 2: Descriptive Statistics After Common Log Transformation

	N		M		SD	
Prob	H	C	H	C	H	C
1.1	104	93	2.63	2.79	0.48	0.55
1.2	88	76	2.59	2.73	0.54	0.54
1.3	90	67	2.62	2.72	0.44	0.48
1.4	87	71	2.66	2.89	0.40	0.41
1.5	84	67	2.55	2.75	0.48	0.60

To test for differences between the two groups on each problem, we subjected the common log transformed data to t-test. The results from this test are presented in table 3. There are significant differences for problems 1, 4, and 5. The ratio is calculated by taking the difference between the hint group mean and the control group mean. As $\lg(x) - \lg(y) = \lg(\frac{x}{y})$ the confidence interval from the logged data estimates the difference between the population means of log transformed data. Therefore, the anti-logarithms of the confidence interval provide the confidence interval for the ratio. We provide the C:H ratios and confidence intervals in table 4.

Table 3: Ratio Between Groups (H:C) in the Original Scale

		95% Confidence Interval			
Prob	Ratio	low	high	p-value	t
1.1	0.69	0.50	0.97	0.03	-2.18
1.2	0.72	0.49	1.06	0.10	-1.68
1.3	0.78	0.56	1.10	0.15	-1.43
1.4	0.58	0.44	0.78	0.00	-3.61
1.5	0.62	0.42	0.93	0.02	-2.31

In order to explore what these differences mean, we shall transform the data back to our original scale (seconds.) The transformed data is provided in table 5. These are the Geometric Means, which are often closer to the original median, than they are the mean. The ratios from tables 3 and 4 are easily interpreted as the log of the ratio of the geometric means. For example in problem 1.4, in the common log scale, the mean difference between hint and control group is -0.23. Therefore, our best estimate of the ratio of the hint time and control time is $10^{-.23} = 0.58$. Our best estimate of the effect of Hint is it takes 0.58 times as many seconds as the control group to complete the problem. The confidence interval reported above is for this difference ratio.

Table 4: Ratio Between Groups (C:H) in the Original Scale

		95% CI	
Prob	Ratio	low	high
1.1	1.44	1.04	2.01
1.2	1.39	0.94	2.05
1.3	1.27	0.91	1.78
1.4	1.71	1.28	2.30
1.5	1.60	1.07	2.40

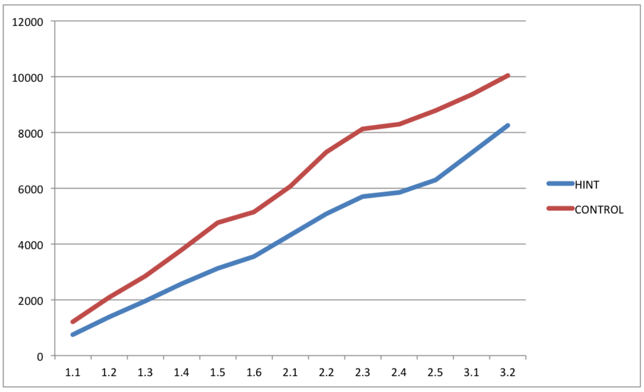


Figure 1: Cumulative average time (in seconds) per problem across the tutor.

The geometric mean of the amount of seconds needed to solve problem 1.4 for the hint group is 0.58 (95% CI: 0.44 to 0.78) times as much as that needed for students in the control group. Stated alternatively, students in the control group spend 1.71 (95% CI: 1.07 to 2.40) times as long as the Hint group in problem 1.4.

Table 5: Geometric Means and Confidence Intervals in Seconds

		95% CI			95% CI	
P	H	low	high	C	low	high
1	428.66	347.14	529.31	618.19	478.60	798.51
2	387.07	297.97	502.82	537.80	405.75	712.82
3	413.80	335.89	509.78	527.18	405.05	686.13
4	454.43	374.38	551.61	778.01	624.48	969.29
5	352.90	278.06	447.89	565.61	405.34	789.24

Exploring the total time spent between all five problems also required a log transformation. The total time spent on the first 5 problems between the hint group (M = 3.34, SD = 0.4) and the control group (M = 3.44, SD = 0.51) was not significant, $t(198) = 1.41, p = 0.16$. This corresponds to a H:C ratio of 0.81 (95% 0.60 to 1.09), and a C:H ratio of 1.24 (95% CI: 0.92 to 1.66).

In order to explore differences in overall time in tutor between the two groups, we subjected the total elapsed time on all 13 problems. The total time in tutor between the hint group (M = 3.75, SD = 0.43) and the control group (M = 3.72, SD = 0.58) was no significant, $t(200) = 0.40, p = 0.694$.

3.1 Hint Usage and Coverage

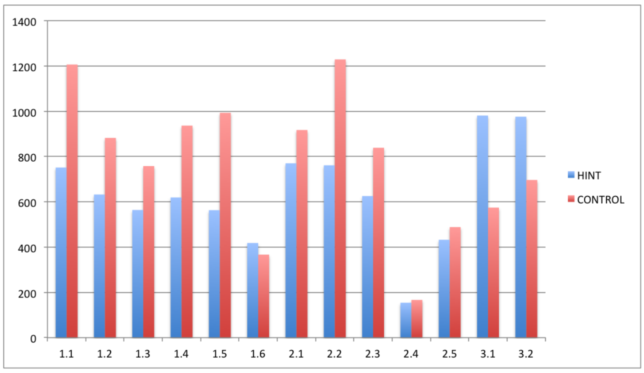


Figure 2: Average time (in seconds) spent per problem.

We investigated the average hint usage per student, per problem. Table 6 depicts the average number of hints per student for each hint level, for each problem. Note that these values are for a single problem, which requires multiple steps. This means that requesting a level four hint allows a student to skip a single step, of many, for a single problem and not an entire problem.

Table 6: Average Hint Use per Problem

Problem	H1	H2	H3	H4
1.1	1.61	0.94	0.66	0.23
1.3	1.79	1.46	1.13	0.77
1.4	2.96	1.66	1.18	0.32
2.2	3.44	2.27	2.04	1.08
2.3	5.56	3.09	2.44	1.00
2.4	1.45	0.99	0.90	0.51
2.5	3.66	1.91	1.66	0.88

In table 7 we provide the hint coverage for each problem. The hint coverage is calculated by taking the number of fulfilled hint requests divided by the number of total hint requests for a problem.

Table 7: Hint Coverage Rates

Problem	Hint Coverage
1.1	0.74
1.3	0.62
1.4	0.81
2.2	0.82
2.3	0.81
2.4	0.88
2.5	0.80
Overall	0.78

4. DISCUSSION

The results of this analysis show that students in the control group are overall not spending significantly more time in the tutor during these first five problems. However, the control does spend significantly more time in some problems compared to the hint group. Problems 1, 3 and 4 provided students with the automatically generated hints. While problem 2 and 5 had no hints for either group. We would expect there to be differences in time to solve for the hint group, and this was the case for problem 1. We would also expect that having no hints on problem two would not display an effect, as the second problem is too early to expect differences to emerge between the groups. Problem 1.3 is interesting as this problem is the first in which the groups begin to show preferences towards different solution strategies. With the control group preferring to work backwards, and the hint group preferring to work forwards (hints are only available for solutions working forward). Problem 1.4 and 1.5, both of which showed significant differences in time spent, showed a large portion of control group student interactions to be perusing buggy-strategies.

This is interesting as the control group is spending at least as much, and often more, time in tutor and yet meeting with less overall success. The control students are not becoming stuck in a single bottleneck location within the problems and then quitting, which would result in lower control group times. The control students are actively trying to solve the problems using strategies that do not work. The hint group is able to avoid these strategies via the use of the hints. The hint group students also develop a preference for solving problems forward, as that is the direction in which they can ask for hints. It is interesting to see that these preferences remain, even when hints are not available.

The effect of the automatically generated hints appear to let the hint group spend around 60% of the time per problem compared to the control group. Or stated differently, the control group requires about 1.5 times as much time per problem when compared to the hint group. These results show that the hints provided by the Hint Factory, which are generated automatically, can provide large differences in how long students need to solve problems.

Regarding average hint use, table 6 suggests that problem 2.3 is likely the most difficult as it has the highest levels of hint usage for nearly all levels. Table 6 also suggests there is little gaming behavior occurring in the Deep Thought tutor from students. As previously stated a single problem requires multiple steps, so to see level four hints at values around one and below is encouraging.

5. CONCLUSIONS AND FUTURE WORK

This paper has provided evidence that automatically produced hints can have drastic effects on the amount of time that students spend solving problems in a tutor. We found a consistent trend in which students without hints spent more time on problems when compared to students that were provided hints. Exploration of the interaction networks for these problems revealed that the control group often spent this extra time pursuing buggy-strategies that did not lead to solutions. Future work will explore other data available on the interaction level, such as errors, in order to get a better understanding of what the control group is doing with their extra time in tutor. We will also look into the development of further interventions that can help students avoid spending time on strategies that are unlikely to provide solutions.

6. REFERENCES

[1] T. Barnes and J. Stamper. Toward automatic hint generation for logic proof tutoring using historical student data. *Proceedings of the 9th International Conference on Intelligent Tutoring Systems (ITS 2008)*, pages 373–382, 2008.

[2] B. S. Bloom. *Taxonomy of Educational Objectives: The Classification of Educational Goals.* Taxonomy of educational objectives: the classification of educational goals. Longman Group, New York, 1956.

[3] M. J. Croy. Graphic interface design and deductive proof construction. *J. Comput. Math. Sci. Teach.*, 18:371–385, December 1999.

[4] M. J. Croy. Problem solving, working backwards, and graphic proof representation. *Teaching Philosophy*, 23:169–188, 2000.

[5] M. Eagle, M. Johnson, and T. Barnes. Interaction Networks: Generating High Level Hints Based on Network Community Clustering. *educationaldatamining.org*, pages 1–4.

[6] D. Fossati, B. Di Eugenio, S. Ohlsson, C. Brown, L. Chen, and D. Cosejo. I learn from you, you learn from me: How to make ilist learn from students. In *Proceedings of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*, pages 491–498, Amsterdam, The Netherlands, The Netherlands, 2009. IOS Press.

[7] T. Murray. Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education (IJAIED)*, 10:98–129, 1999.

[8] J. Stamper, T. Barnes, L. Lehmann, and M. Croy. A pilot study on logic proof tutoring using hints generated from historical student data. *Proceedings of the 1st International Conference on Educational Data Mining (EDM 2008)*, pages 197–201, 2008.

[9] J. C. Stamper, M. Eagle, T. Barnes, and M. Croy. Experimental evaluation of automatic hint generation for a logic tutor. In *Proceedings of the 15th international conference on Artificial intelligence in education*, AIED'11, pages 345–352, Berlin, Heidelberg, 2011. Springer-Verlag.

[10] K. VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.

Providing implicit formative feedback by combining self-generated and instructional explanations

Joseph Jay Williams[1]
[1]University of California at Berkeley
joseph_williams@berkeley.edu

Helen Poldsam
Talinn Tech
hpoldsam@gmail.com

Abstract. Formative feedback for a learner typically uses human or artificial intelligence to draw an inference about a learner's knowledge state from the learner's actions, and select a learner-directed response. To tackle cases when such intelligence is not easily available, we are exploring ways of providing implicit formative feedback: A learner's action is to respond to an explanation prompt, and the learner-directed response is to provide an instructional explanation. We consider explanations for correct examples to mathematics exercises, but the exciting implications will be for less well-defined domains that are challenging for cognitive tutors to model. To motivate learners to explain and to increase implicit feedback, we also explore prompts to compare the self-generated and instructional explanations.

Keywords: explanation, self-explanation, learning, comparison, formative feedback

1 Introduction

Traditionally, feedback to students has often been *summative*, such as midterm scores and state exams, where even the application of advanced psychometric techniques leads to measures that provide a summary assessment of some attribute. The pen- and paper- tests typically administered and the time needed for another human to grade and assess places a natural delay between a student's behavior(s) and the provision of feedback about that behavior.

Now that learners' behavior is increasingly in a computerized or online environment, there are three key implications. The first is that many tests and measures typically considered as (summative) assessments can be analyzed instantaneously and automatically. The second is that online digital environments allow for the *delivery* of sophisticated instruction and formative feedback. The third is that the constant logging of data on a computer means that a much wider range of student behaviors is available as fodder for 'assessments', which can then be analyzed and used to provide *formative feedback* to students.

As evidenced by the current workshop and extensive research in the learning sciences [1] [2] [3], great progress has been made in developing formative assessments and feedback. However, the issue of providing formative feedback raises two core challenges.

The first is that providing formative feedback that helps learning seems to be constrained by how accurately an automatic system can diagnose a learner's knowledge state, infer what instructional tactic is likely to deliver formative feedback that moves the learner to a more effective knowledge state, and ensure the learner successfully uses this instruction or formative feedback. While there have been great strides in developing the data mining and artificial intelligence capacities to achieve all three of these goals, is there a way to mitigate these constraints through a complementary approach to the problem of providing formative feedback?

The second challenge is that – even if the above issues could be solved – learners may not learn general metacognitive skills of self-regulation – to identify gaps in their knowledge, consider how to fill them or seek out new information, and engage in effective learning strategies that move their understanding forward.

One potential way to address both of these issues is to provide information from which *learners* can generate *implicit* formative feedback, and structure the instructional environment to support learners in generating and using this feedback.

This paper outlines a paradigm for doing this and reports the design of an ongoing study. Learners are asked or prompted to self-generate explanations, then are provided with normative answers or instructional explanations that respond to the same prompt, and finally are guided to compare their self-generated explanations with the instructional explanations provided. This draws together work in education and psychology on the benefits of *self-explanation* [4] [5], on how to provide appropriate instructional explanations for students [6] [7], and on the benefits of comparison for learning [8].

Context for introducing implicit feedback: Worked example solutions in Khan Academy mathematics exercises

While the general framework can be applied to many contexts, the current study examines the generation, consideration, and comparison of explanations in the KhanAcademy.org exercise framework (www.khanacademy.org/exercisedashboard). This provides a large collection of mathematics exercises with a similar format, used by tens of thousands of students. It is therefore a widely applicable context in which to develop a paradigm for providing implicit feedback from self-generated and instructional explanations.

Figure 1 shows an example of an exercise we have augmented. The typical (non-augmented) exercise starts with a statement of a problem for the student to solve, which is outlined in the box surrounded by a dark black line. When ready, students can type in a proposed answer and then receive feedback on its correctness. At any point, students can also request a hint, which reveals the next step of a worked example solution to the problem. Students have to enter the correct problem to advance, but because every problem provides on-demand "hints" which step-by-step reveal the solution, they can eventually do so (the last step is simply the answer).

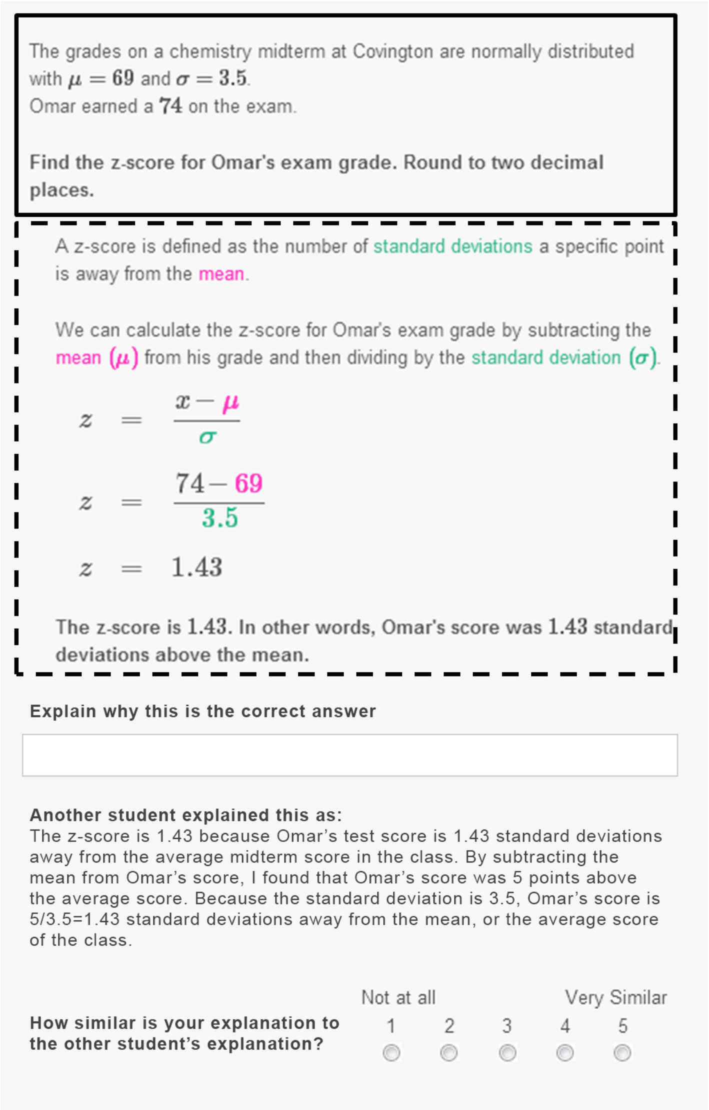
This design already builds in some form of implicit feedback, if it is assumed that students first try to consider steps in the problem's solution before requesting hints. A hint or solution step can therefore give them implicit feedback about the appropriateness of what they were considering before.

Incorporating self-generated and instructional explanations

The template for Khan Academy's mathematics exercises ensures that students must generate or simply be told the correct answer by the end of each exercise. Our augmentation of the exercises all occurs after the student receives feedback that they have entered the correct answer – whether they generate it themselves, are helped by hints, or need to go to the very end of the solution to see the answer.

As shown in Figure 1, the typical Khan Academy math exercise (labeled *practice-as-usual*) is augmented using three instructional tactics: (1) Including prompts for students to *self-generate* explanations; (2) Including *instructional* explanations directed at these prompts, ostensibly from another student or teacher; (3) Asking students to *compare* their self-generated explanations to the instructional explanations.

The *self-generate* explanation prompt appears beside a solution step, in a distinctive purple font and accompanied by a text box for students to type their response. The example in Figure 1 has the prompt "Explain what this step means to you:". The *instructional* explanation can be shown in a similar position, such as "Another student explained this as:…". The *compare* judgment solicits a comparison of the student's own explanation with the *instructional* explanation which was supposedly provided by someone else: "How similar is your explanation to the other student's explanation?".



The grades on a chemistry midterm at Covington are normally distributed with $\mu = 69$ and $\sigma = 3.5$.
Omar earned a 74 on the exam.

Find the z-score for Omar's exam grade. Round to two decimal places.

A z-score is defined as the number of standard deviations a specific point is away from the mean.

We can calculate the z-score for Omar's exam grade by subtracting the mean (μ) from his grade and then dividing by the standard deviation (σ).

$$z = \frac{x - \mu}{\sigma}$$

$$z = \frac{74 - 69}{3.5}$$

$$z = 1.43$$

The z-score is 1.43. In other words, Omar's score was 1.43 standard deviations above the mean.

Explain why this is the correct answer

Another student explained this as:
The z-score is 1.43 because Omar's test score is 1.43 standard deviations away from the average midterm score in the class. By subtracting the mean from Omar's score, I found that Omar's score was 5 points above the average score. Because the standard deviation is 3.5, Omar's score is 5/3.5=1.43 standard deviations away from the mean, or the average score of the class.

	Not at all				Very Similar
	1	2	3	4	5
How similar is your explanation to the other student's explanation?	○	○	○	○	○

Figure 1: Illustration of worked example solution in typical Khan Academy exercise, and how the problem can be augmented with: (1) a prompt to self-generate an explanation for the correct answer, (2) An instructional explanation, ostensibly from another student, (3) A request for a learner to compare his/her explanation with the instructional explanation. The *practice-as-usual* exercise can be found at https://www.khanacademy.org/math/probability/statistics-inferential/normal_distribution/e/z_scores_1

Experiment

The ongoing study will be conducted using a convenience sample of adults recruited from Amazon Mechanical Turk, as well as undergraduate students. The goal is to investigate this paradigm in a controlled laboratory setting, and then extend it to a realistic educational environment with students in a high school, or introduce it on the actual Khan Academy platform, in an extension of an ongoing collaboration with Khan Academy.

The study independently manipulates whether or not learners are prompted to *self-generate* explanations for the correct answer (once it is obtained), and whether or not they are provided an *instructional* explanation for the correct answer. This results in four conditions:

Practice-as-usual with the typical Khan Academy exercise and no self-generated or instructional explanation.

Self-generated explanation (but no instructional explanation) which includes the prompt to explain why the answer is correct.

Instructional explanation (but no prompt to self-generate an explanation) which provides an explanation that is supposed to come from another student.

Self-generated and *instructional* explanations. This condition is key to evaluating whether learning can be improved through using explanations to provide implicit formative feedback for learners. As described in the next section, several variables are manipulated in this condition to investigate the most effective means of combining self-generated and instructional explanations.

Self-generated and instructional explanations: Order & Comparison

To further investigate the learning benefits of self-generated and instructional explanations, the condition in which participants receive both a self-generated and instructional explanation is made of four nested conditions. These are generated by experimentally manipulating the *order* of self-generated and instructional explanation (self-generated prompt first, then instructional explanation, vs. instructional then self-generated) and whether or not a *comparison* is requested (no comparison prompt, vs. a comparison prompt). The comparison prompt asks learners to rate similarity of self-generated and instructional explanations, such as can be seen in Figure 1: “How similar is your explanation to the other student’s explanation?”, rated on a scale from 1 (not at all) to 5 (very similar).

It should be noted that the self-generated and instructional explanation are never onscreen at the same time, to avoid simple copying or rote responses. Whichever is presented first simply disappears on the appearance of whichever is presented second.

The design therefore produces four conditions: *Self-Instructional*, *Instructional-Self*, *Self-Instructional-Compare*, *Instructional-Self-Compare*. The manipulations that produce these conditions allow us to investigate whether and when learners receive implicit formative feedback from generating explanations, receiving instructional explanations, and engaging with prompts to compare these explanations.

Summary

The study outlined here aims to investigate whether the proposed combinations of self-generated and instructional explanations have a beneficial impact on learning. The study can shed light on how to design a learning environment to provide implicit formative feedback, by examining how accuracy and speed in exercises is influenced by the relative effects of self-generating explanations, receiving instructional explanations, doing both, and comparing one's self-generated effort with an instructional explanation. More generally, the software adaptation of the Khan Academy exercise framework provides a setting to ask an even broader range of issues: such as changing the type of explanation prompts, features of the instructional explanations, the kinds of comparison prompts used (listing vs. rating, analyzing differences vs. similarities, contrasting explanation quality by identifying pros & cons of each, or by grading or rating different explanations).

References

1. Nicol, D. J., & Macfarlane‐Dick, D. (2006). Formative assessment and self‐regulated learning: A model and seven principles of good feedback practice. *Studies in higher education*, 31(2), 199-218.
2. Gikandi, J. W., Morrow, D., & Davis, N. E. (2011). Online formative assessment in higher education: A review of the literature. Computers & Education, 57(4), 2333-2351.
3. Shute, V. J. (2008). Focus on formative feedback. Review of educational research, 78(1), 153-189.
4. Fonseca, B. & Chi, M.T.H. 2011. The self-explanation effect: A constructive learning activity. In: Mayer, R. & Alexander, P. (Eds.), *The Handbook of Research on Learning and Instruction* (pp. 270-321). New York, USA: Routledge Press.
5. Williams, J. J., Walker, C. M., Lombrozo, T.: Explaining increases belief revision in the face of (many) anomalies. In: N. Miyake, D. Peebles, & R. P. Cooper (Eds.), Proceedings of the 34th Annual Conference of the Cognitive Science Society (pp. 1149-1154). Austin, TX: Cognitive Science Society (2012)
6. Wittwer, J. & Renkl, A. (2008). Why instructional explanations often do not work: a framework for understanding the effectiveness of instructional explanations. *Educational Psychologist, 43*, 49-64.
7. Renkl, A. (2002). Worked-out examples: Instructional explanations support learning by self-explanations. Learning and instruction, 12(5), 529-556.
8. Gentner, D., Loewenstein, J., & Thompson, L. (2003). Learning and transfer: A general role for analogical encoding. Journal of Educational Psychology, 95(2), 393-408.

An architecture for identifying and using effective learning behavior to help students manage learning

Paul Salvador Inventado[*] , Roberto Legaspi, Koichi Moriyama, Ken-ichi Fukui and Masayuki Numao
The Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka, Japan, Osaka, 567-0047
{inventado,roberto,koichi,fukui}@ai.sanken.osaka-u.ac.jp, numao@sanken.osaka-u.ac.jp

ABSTRACT

Self-regulated learners are successful because of their ability to select learning strategies, monitor their learning outcomes and adapt them accordingly. However, it is not easy to measure the outcomes of a learning strategy especially while learning. We present an architecture that allows students to gauge the effectiveness of learning behavior after the learning episode by using an interface that helps them recall what transpired during the learning episode more accurately. After an annotation process, the profit sharing algorithm is used for creating learning policies based on students' learning behavior and their evaluations of the learning episode's outcomes. A learning policy contains rules which describe the effectiveness of performing actions in a particular state. Learning policies are utilized for generating feedback that informs students about which actions could be changed or retained so that they can better adapt their behavior in future learning episodes. The algorithms were also tested using previously collected learning behavior data. Results showed that the approaches are capable of building a logical learning policy and utilize the policy for generating appropriate feedback.

Keywords

delayed feedback, self-regulated learning, profit sharing

1. INTRODUCTION

Students often learn on their own when they study for tests, make assignments and perform research as part of their academic requirements. They also learn by themselves when they investigate topics which may not be directly related to class discussions but are interesting to them. When students learn alone, they encounter many challenges related to the learning task, as well as challenges that are meta-cognitive and affect related.

Students who can self-regulate are capable of overcoming these challenges better compared to those who cannot. One reason for this is that self-regulated students know how to select and adapt their learning strategies depending on the current situation. However, this is a complex task because it requires attention and sophisticated reasoning to know which learning strategies to apply, to monitor the outcomes of a learning strategy and to know when a strategy needs to be changed [13].

In this research, we discuss an architecture for helping students manage their learning behavior by helping them become aware of the outcomes of the learning strategies they employed and by helping them identify which strategy is effective in a particular situation.

2. RELATED WORK

Self-regulated learners can be differentiated from less self-regulated learners by looking at the learning behaviors they exhibit. They are characterized by their diligence and resourcefulness, their awareness of the skills they possess, their initiative to seek out information and their perseverance to continue learning and find ways to overcome obstacles [13].

Research such as that of Kinnenbrew, Loretz and Biswas [8] has shown these differences in behavior. In their work, they investigated students' learning behavior while using Betty's Brain, a computer-based learning environment in the science domain that helped students develop learning strategies. They processed log data from student interactions and mapped them to canonical actions. Action sequences were then mined using sequential pattern mining and episode mining to discover learning behaviors. Their results showed that high performing students showed systematic reading behavior and frequent re-reading of relevant information which was not seen in low performing students.

In the work of Sabourin, Shores, Mott and Lester [10], the authors also observed differences in the students' behavior as they interacted with Crystal Island, a game-based learning environment developed for the microbiology domain. While interacting with the environment, students were prompted to report their mood and status. These were later processed and used to categorize the students' goal setting and goal

[*] also affiliated with: Center for Empathic Human-Computer Interactions, College of Computer Studies, De La Salle University, Manila, Philippines

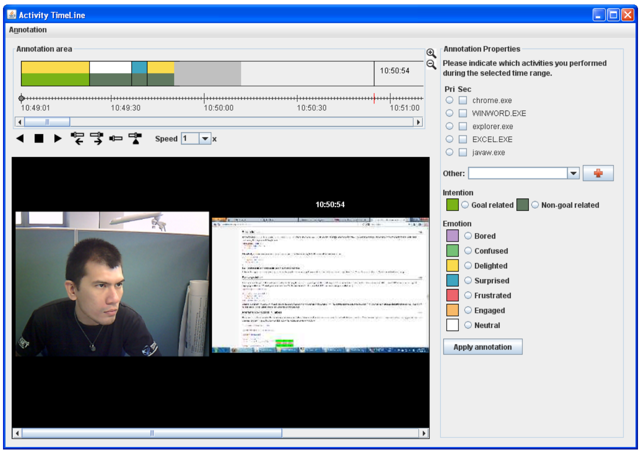
reflection behavior. They were then given an overall self-regulated learning (SRL) score based on their reports and assigned into low, medium or high SRL category. Students in the high SRL category frequently used in-game resources that provided task-related information and resources that allowed them to record notes. They also spent less time using resources for testing their hypothesis and had higher learning gains.

MetaTutor is a hypermedia learning environment developed for the biology domain that identifies students' SRL processes and also helps them use these processes [2]. Students who used the system indicated the SRL processes they used by selecting it from the list of SRL processes in the system's interface. Pedagogical agents also gave them prompts to use certain SRL processes depending on the current situation (i.e., student information, time on page, time on current sub-goal, number of pages visited relevance of the current page to the sub-goal, etc.) and also gave them feedback regarding how they used these processes. Students who used the version of the system with prompts and feedback were reported to have higher learning efficiencies compared to students who used a version of the system without prompts and feedback.

3. SYSTEM ARCHITECTURE

Learners often have difficulty in selecting, monitoring and adapting learning strategies because of its high cognitive load requirement. This is especially true for complex domains such as science, math, engineering and technology. The approach we take in this work involves helping students understand the outcomes of their learning behavior better by helping them recall what transpired in a recently concluded learning episode. The advantage of recalling is that after the learning episode, students do not need to worry about the learning task and can focus on analyzing their learning behavior. Students will also have a more complete and accurate measurement of their learning behavior's effectiveness because they can observe both short and long term effects on learning. This information will be useful for students in future learning episodes because when they monitor and adapt learning strategies, they can base their decisions on the current context as well as their predictions of what could happen according to their reflections from previous learning episodes.

Asking students to recall a recently concluded learning episode presents two issues. First, students will not be able to completely remember what transpired during the learning episode. We addressed this in our previous work wherein we developed a tool called Sidekick Retrospect, which took screenshots of the students' desktop and video frames from a video of their face during a learning episode [7]. Students who used the software in our experiment reported that they were able to discover things about their behavior that they were previously unaware of. It was also enough to help them reflect on what transpired so that they were able to identify problems with their learning behavior and think of probable solutions. Figure 1 shows a screenshot of the system's interface which are presented to the students after the learning episode. A timeline of the entire learning episode is shown together with desktop and webcam video screenshots relative to the mouse's position in the timeline.



Please indicate which activities you performed during the selected time range.

Speed 1 x

Figure 1: Sidekick Retrospect Annotation Interface

An issue we encountered from our previous work was that students who used the software seemed to focus only on the most significant aspect of the learning episode. They did not reflect as much on other instances during the learning episode even when they employed other learning strategies that also had an impact on their learning. This may have been the case because students were already too tired to spend more time analyzing each event in depth.

The architecture presented in Figure 2, integrates the methodology we used in our previous work with our current approach for helping students recall what transpired during the learning session and helping them discover more insights about their learning behavior. We designed our system so that students would not be bound by a specific environment or domain and keep the learning environment as natural as possible. Students were allowed to learn using any tool or application on or off the computer. However, they had to stay in front of the computer so it could take desktop and webcam video screenshots of their activities and so they could annotate the data after the learning episode. The entire process was split into three phases which are each discussed in the following subsections.

3.1 Interaction Phase

The interaction phase begins by first asking students to input their learning goals for the current learning episode. Data collection starts right after students finish inputting their goals. The system then starts logging the applications used by the students, the title of the current application's window and the corresponding timestamps. Screenshots of the desktop and the webcam's video feed are also taken and stored using the same timestamp as that of the log data.

3.2 Annotation Phase

In the annotation phase, students are asked to annotate their *intentions*, *activities* and *affective states*. Intentions can either be goal related or non-goal related relative to the goals that were set at the start of the learning episode. Activities referred to any activity the student did while learning which could either be done on the computer (e.g., using a browser) or out of the computer (e.g., reading a book). Two sets of affect labels were used for annotating affective states

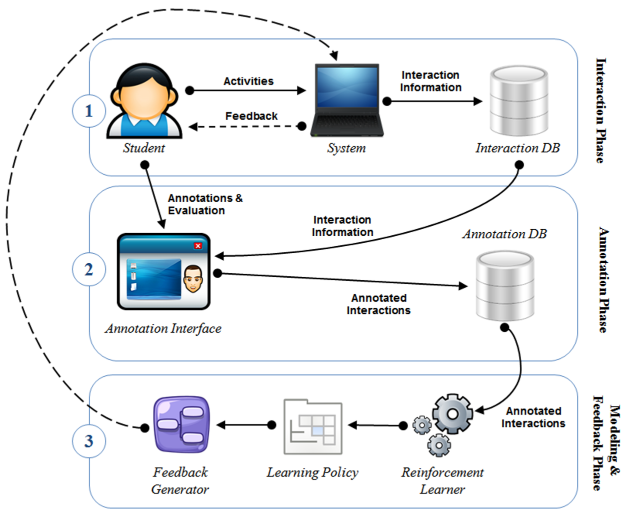


Figure 2: System architecture

wherein goal-related activities were annotated as either delighted (DEL), engaged (ENG), confused (CNF), frustrated (FRS), bored (BRD), surprised (SRP) or neutral (NUT) and non-goal related activities were annotated as either delighted (DEL), sad (SAD), angry (AGY), disgusted (DIS), surprised (SRP), afraid (AFR) or neutral (NUT). Academic emotions [4] are used for annotating goal related intentions because they give more contextual information about the learning activity. However, academic emotions might not capture other emotions outside of the learning context so Ekman's basic emotions [5] were used to annotate non-goal related intentions.

The system's annotation interface helps students recall what transpired during the learning episode by showing desktop and webcam screenshots depending on the position of the mouse on the timeline. The actual annotations can be created by using the mouse to select a time range then clicking on the corresponding intention, activity and affective state buttons. Students are also allowed to input a description of the activity when it was done outside of the computer.

While annotating, students inherently recall what transpired allowing them to identify the appropriate annotation. Going through the entire learning episode sequentially also helps the students annotate more accurately because they can see how and why their activities change. Furthermore, they also see the outcomes of these activities. It is possible that students might not annotate the data correctly for fear of judgment or lower scores. However, reassuring them that the results will not be used as part of their grades or telling them that accurately annotating their data will help them become more self-regulated and effective could help minimize these cases.

After the annotation process, students are asked to give a learning effectiveness rating between one to five, indicating how good they felt the learning episode was. This rating is likely to be accurate because of the level of detail in which students reviewed their learning episode.

3.3 Modeling and Feedback Phase

In the modeling and feedback phase, students' data are analyzed to create and update the student's list of effective learning behavior or policy. Students' behavior in the current learning episode can be compared to the policy to identify effective and ineffective behavior that can be adapted in succeeding episodes.

3.3.1 *Learning policy creation*

Self-regulation can be viewed as cyclic phases of forethought, performance and self-reflection [14] wherein reflections about the outcomes of behavior after a learning episode can be used to increase the effectiveness of future learning episodes (e.g., discarding or modifying ineffective behavior). The ideal effect of would be for learning outcomes to continually improve over time.

We fit this incremental perspective of adapting behavior into a reinforcement learning (RL) problem in machine learning which searches for the best actions to take in an environment (i.e., learning behavior) to maximize a cumulative reward (i.e., learning effectiveness) [11].

Profit sharing is a model-free RL approach that is capable of converging even in domains that do not satisfy the Markovian property [1]. We decided to use this approach primarily because we deal with human behavior in a non-deterministic and uncontrolled environment. Profit sharing's reinforcement mechanism allows it to learn effective, yet sometimes non-optimal, policies quickly compared to other algorithms. This is ideal for our situation because we need to give policy-based feedback using minimal data.

Profit sharing differs from other RL techniques because it reinforces effective rules instead of estimating values from succeeding sequential states. A rule consists of a state-action pair (O_t,A_t) which means performing A_t when O_t is observed. We consider these rules as learning behaviors. An episode n is a finite sequence of rules wherein the entire sequence is awarded the reward R based on its outcome. After each episode, the weights of each rule in the sequence is updated using (1) where function $f(R,t)$ is a credit assignment with t being the rule's distance from the goal. Note that it is possible for a rule's weight to be updated more than once if it appears more than once in a sequence. The set of all rules and their corresponding weights is called a policy. A policy is rational or guaranteed to converge to a solution when the credit assignment fuction satisfies the rationality theorem (2) with L being the number of possible actions. In our work, we used a modified version of the rational credit assignment function (3), which was adapted from [1] so that the rules' weights will be bound by the reward value.

$$W_{n+1}(O_t, A_t) \leftarrow W_n(O_t, A_t) + f(R, T) \tag{1}$$

$$\forall t = 1, 2, 3..., T. \quad L \sum_{j=0}^{t} f(R, j) < f(R, t) \tag{2}$$

$$f_{n+1}(R, t) = (R - W_n(O_t, A_t))(0.3)^{T-t} \tag{3}$$

According to Winne's [12] SRL model, students adapt their strategies based on the results of metacognitive monitoring and evaluation. When the outcome of a task satisfies a student's expectations, then they may continue performing the current task or proceed to the next task. On the other hand, when a task does not achieve its expected outcomes, students can adapt their strategies accordingly. Unfortunately, we did not have access to students' metacognitive evaluations in our data. However, Carver and Scheier's [3] model theorized that the results of metacognitive evaluations can be observed in students' emotion. When the outcome of a task is according to a student's expectation, then neutral affect is experienced. However, when the outcome does not satisfy expectations then negative affect is experienced. On the other hand, when the outcome exceeds expectations then positive affect is experienced. Based on these assumptions, we represented our states using the triple <activity, affect, duration>. Apart from affect which approximated students' metacognitive evaluation, we included activity to indicate the task performed by the student and duration to indicate how long it was performed by the student.

The data showed that students performed similar activities but used different applications (e.g., browsing websites with Google Chrome vs. Mozilla Firefox). Instead of treating these separately, we categorized the students' activities into six types: information search [IS] (e.g., using a search engine), view information source [IV] (e.g., reading a book, viewing a website), write notes [WN], seek help from peers [HS] (e.g., talking to a friend), knowledge application [KA] (e.g., paper writing, presentation creation, data processing) and off-task [OT] (e.g., playing a game).

Durations were even more varied ranging from one second (e.g., clicking a link from a search results page) to 53 minutes (e.g., watching a video). Using this directly will result in a large state space so we categorized them into short, medium or long duration. The duration values were positively skewed so evenly partitioning the data according to the number of elements or frequency would cause both short and medium groups to have small and similar values. The long duration group on the other hand, would have values with high variation. We decided to use k-Means to categorize the duration values into three clusters (i.e., $k = 3$) and using a Euclidean distance formula as described in [6]. Clustering produced groups with elements having similar duration values and whose values were different from the other groups. Elements in the cluster with the smallest values were labeled short duration, elements in the cluster with the biggest values were labeled long duration and the elements in the remaining cluster were labeled with medium duration. The centroids identified by k-means for short, medium and long durations were 69.4 seconds (1.15 minutes), 614.5 (10.2 minutes) seconds and 1999.4 seconds (33.3 minutes) respectively. 90.83% of the duration values were short, 8.17% were medium and 0.10% were long.

In the learning context, actions would refer to changing from one activity to the other. So, we used the same eight activity categories as actions. However, we added a change information source [CS] action to handle cases when students would either view a different website or change to or from a physical information source (e.g., book, printed conference paper). In this representation, there would be no consecutive rules with states having the same values unless they were paired with different actions. Otherwise, these rules were merged and their durations added. An example of a rule would have the form (<IV, CNF, short>, CS).

The student's rating of the learning episode's effectiveness can directly be used as the reward value. Data from learning episodes can then be converted into rule sequences and be used to update each rule's weight incrementally using (1) with the corresponding reward values. The rules' weights are expected to converge to the reward value it is commonly associated with.

3.3.2 Learning policy-based feedback

According to Pressley, Levin and Ghatala [9], adult students who were given information regarding the utility of two learning strategies and a chance to practice them were capable of validating its outcomes and were reported to use the more effective strategy. In our case, the utility of performing an action in a certain state is its weight value (i.e., applying the rule will likely lead to a learning effectiveness rating that is at least the weight value). Information about the utility of two or more competing rules (i.e., rules referring to the same state but with different actions) can be used to give students feedback at the end of a learning episode so they can verify and adapt them accordingly in succeeding episodes. When students used more effective rules, it is assumed to result in better learning effectiveness ratings which will reinforce the rule in the learning policy.

As more rules are observed and added into the learning policy, some rules may not be relevant to a particular learning episode. The rules with their corresponding utilities should first be filtered before they are presented to the student. In the first learning episode, the learning policy will still be empty so feedback will be unavailable. When a policy already contains rules, each rule employed in the current learning episode can be compared to the rules in the learning policy and provide relevant feedback. The pseudo code presented below describes how three types of feedback can be given to the student. First, when students perform an action with a worse utility based on the policy, the system can remind the student to select the better action. Second, if the student performs an action which isn't in the policy but has lower utility than the best action in the policy, the student is told that the action may be ineffective. Lastly, if the student performs an action which isn't in the policy but has a higher utility than the best action, the student is informed that a better action has been found compared to the previous best action. Whenever a student performs the best action according to the policy, feedback is no longer given because it is assumed that the student already knows this and is the reason why the action was selected. In cases when the student performs an action in an unknown state, feedback cannot be given as well because of insufficient information.

```
Initialize set of weighted rules $X$
Copy old policy P into P'
For each $(O_t, A_t)$ in the current learning episode
    Update $W(O_t, A_t)$ in P' using (1)
```

```
  For each $(O_p, A_p)$ in policy $P$
    If $O_t = O_{p,i}$
      Add $W(O_{p,i}, A_{p,i})$ into $X$
    End
  End
End
For each $(O_t, A_t)$ in the current learning episode
  If $(O_t, A_t)$ not in $X$
    Unknown utility
  Else if $(O_t, A_t)$ not in $P$
    If $W(O_t, A_t) < max(W(O_{p,i'}, A_{p,i'}))$ in $X$
      Inform student that $A_{p,i'} > A_t$
    Else
      Inform student that $A_t > A_{p,i'}$
    End
  Else
    If $A_t <> A_{p,i'}$ where $max(W(O_{p,i'}, A_{p,i'}))$ in $X$
      Inform student that $A_{p,i'} > A_t$
    End
  End
End
```

A cause for concern is that the learning policy might not have converged yet resulting in incorrect feedback (e.g., telling the student to perform an action which is actually ineffective). Again according to Pressley *et. al.* [9], despite being given incorrect utility information adults are able to select better strategies after practice wherein they are able to observe the strategy's actual utility. As students constantly select effective actions (i.e., as a result of their own evaluation), the policy will be updated to reinforce these actions and decrease the chance of providing incorrect feedback. This emphasizes the need for students in this environment to explore other actions so that they can find the best actions which will also be reflected in the policy. It also then becomes necessary for other mechanisms to encourage exploration such as looking at other students' learning policies for possible actions or using expert knowledge.

4. LEARNING BEHAVIOR DATA

The methodology described in the interaction and annotation phases of the architecture was used in collecting the data in our previous work [7]. The data was collected from four students aged between 17 and 30 years old, conducting research as part of their academic requirements. Three of the students were taking Information Science while one student was taking Physics. During the data collection period, two of the students were writing conference papers and two made power point presentations about their research. They all processed and performed experiments on their collected data, searched for related literature and created a report or document. Although their topics were different, they performed similar types of activities. Two hours of annotated learning behavior data in five separate learning episodes were collected from each student over a one week period. The annotation data was processed using the method described in Section 3.3.1 resulting in five separate learning episodes for every student and each episode consisting of the sequenced rules. On average, students used 54.35 rules per session (N=20; σ=27.71) including repeated rules.

Table 1: Rule Categories

#	Type	State	Action	Reward
1	PRL	ENG, IV, short	KA	0.360000
2	PRL	ENG, IV, short	CS	0.004154
3	CDH	CON, IV, short	CS	0.441939
4	CDH	CON, IV, short	KA	2.34E-05
5	CDH	CON, IV, short	OT	9.16E-15
6	RLX	ENG, KA, long,	OT	1.830000
7	RLX	ENG, KA, long,	HS	0.009720
8	RLX	ENG, KA, long,	IV	2.13E-06
9	RSL	DEL, OT, short	KA	0.389484
10	RSL	DEL, OT, short	IV	2.00E-18
11	RSL	DEL, OT, short	HS	9.57E-26

5. RESULTS AND ANALYSIS

The learning policies generated by the profit sharing algorithm on the learning behavior data consisted of rules based on the state and action representation used. There were many rules due to our selected state-action space, but we observed four categories after analyzing the data– Prolonged learning (PRL), Cognitive disequilibrium handling (CDH), Relaxation (RLX) and Resumed learning (RSL). Table 1 presents examples of each category which were taken from the learning policy of the doctoral physics student who was experimenting with her data and used its results for writing a conference paper.

PRL rules refer to states wherein students feel engaged while performing a learning-related activity and switch to another learning-related activity. It describes how long a certain type of activity could be effective and what other activities may complement it. Taking the physics student's data as an example, let us consider that she was looking into different concepts for data manipulation because she needed it for writing her conference paper. According to rules 1 and 2, it was better for her to try and run an experiment on her data (i.e., apply knowledge), before shifting to a different concept (i.e., view information source). This would allow her to have a better understanding of the concept and allow her to write the paper more easily.

CDH rules refer to states wherein students adapt their behavior to handle negative affect (e.g., confusion or boredom) while learning. These give an idea how long to stay in a confusing or bored learning state before shifting to an activity that will probably alleviate the problem. For example, rule 3 indicates that it is probably better to find a different information source if it is confusing instead of spending a lot time trying to understand it. Rule 5 also indicates that it is not a good idea to just engage in off-task activities when it is difficult to understand a certain information source.

RLX rules refer to states wherein students relax or shift to off-task activities after learning. According to rule 6, it was effective for the student to relax after spending a long time learning. This supports claims that off-task activities or relaxation are important for continued learning [7].

RSL rules refer to cases wherein students shift back to learning from an off-task activity. It seemed that the utility for performing actions in this category are context-dependent.

Table 2: Rule correctness over learning episodes

Ep	+	-	New^+	New^-	Unknown	Reward
2	0	0	1	0	3	4
3	1	0	2	1	1	3
4	12	0	5	0	1	4
5	4	51	0	1	6	2

For example, according to rule 9, it was more effective to apply knowledge probably because the goal was to write a conference paper. Spending too much time reading information sources would help, but not directly lead to the achievement of the goal. This effect is important to consider because if students change their goals, the policy may not be directly applicable to the new goal. A separate experiment needs to be conducted to observe how the architecture will handle such scenarios. We think however that the speed in which the algorithm adjusts the learning policy is a good factor that can make it capable of handling such changes.

After a student completes a learning episode, an updated learning policy can now be used to generate feedback. The feedback will be based on five cases: the student chooses the best action according to the policy (+), the student does not choose the best action according to the policy (-), the student tries a new action which has better results than the best action in the policy (New^+), the student tries a new action which has worse results than the best action in the policy (New^-) and the student performs the only action associated to a state in the policy or the student performs an action in an unknown state for the first time such that the policy will not be able to identify if there is a better action (Unknown).

We simulated how feedback would be generated for these five cases by testing the algorithm on data from the same student. The student's actions in the first learning episode were used to build an initial policy. No feedback was generated at this point because learning policy would only contain rules based on the current episode. Feedback for the second episode could now be generated because it can be compared with the learning policy created using data from the first learning episode. The learning policy was updated using data from the second episode, and was used to generate feedback for the third learning episode. This was repeated for all remaining learning episodes. Table 2 presents the number of times each case is encountered as new learning episodes are experienced by the student.

The table shows that the student implemented a few rules in episode two which was caused by the student spending a long time performing an activity. We see that her learning policy was updated with three new rules as well as a new effective action (i.e., performing an off-task activity after spending a long time experimenting with data). The high reward value indicates that the student did well because all actions, including those unknown actions, were effective. This was confirmed by checking her updated learning policy generated in the fifth episode. The unknown actions were in fact the best actions in their corresponding states (i.e., performing an off-task activity after spending some time experimenting with data, resume data experimentation after a short off-task activity and consulting a friend about the experiment after a short off-task activity). The student also performed few actions in the third episode but gave it a smaller reward value probably because she spent too much time talking to a friend even though the other actions were effective (i.e., resuming data experimentation after a short off-task activity and viewing a paper after some time experimenting with data). In the fourth episode, the student constantly performed effective actions and even discovered a new action which probably caused the increase in reward. Finally in the fifth episode, the student performed a lot of ineffective actions which probably caused the big decrease in the reward value. Specifically, as we have discussed earlier, she spent short amounts of time repeatedly viewing different information sources. The policy indicated that it would have been better for her to apply knowledge, which in her context would mean either writing the paper or experimenting with her data. This could in fact be an effective strategy because she could verify and learn more about the concept by applying it rather than moving on to another concept right away.

Our results also showed that there was a relationship between the number of times students correctly followed rules in their learning policy and their learning effectiveness rating. Figure 3 presents graphs corresponding to each student showing this relationship. The learning effectiveness ratings were expressed as ratios relative to the highest rating (i.e., five) and the number of correct actions were expressed as ratios relative to the total number of actions in the learning episode. The trend indicates that the learning policy was able to identify effective actions from the students' behavior such that when the students selected more effective actions (i.e., based on the learning policy), they also had a more effective learning episode. This means that if the student will be able to follow the feedback provided by the system in succeeding learning episodes, it is likely for them to have more effective learning experiences.

6. CONCLUSION AND FUTURE WORK

We have presented an architecture for collecting students' learning behavior data, uncovering effective learning behaviors and using them to help students manage their learning. The approach does not require a specific learning environment so the student's behavior is naturalistic and captures how he/she actually learns. However, it does require students to annotate their data. Annotation is done after learning so it does not require additional cognitive load during the learning episode. Desktop and web cam screenshots can help students recall the context in which they learned and can likely improve annotation accuracy.

The profit sharing algorithm was used for building learning policies that contained rules describing an action's effectiveness in a particular state. Learning policies generated from previous learning episodes can be compared with data from the current learning episode to identify which actions were effective or ineffective and generate feedback accordingly. Feedback about possible improvements can be useful for students to adapt their actions in future learning episodes.

Simulations from actual data showed that updating the learning policy also changed the resulting feedback such that

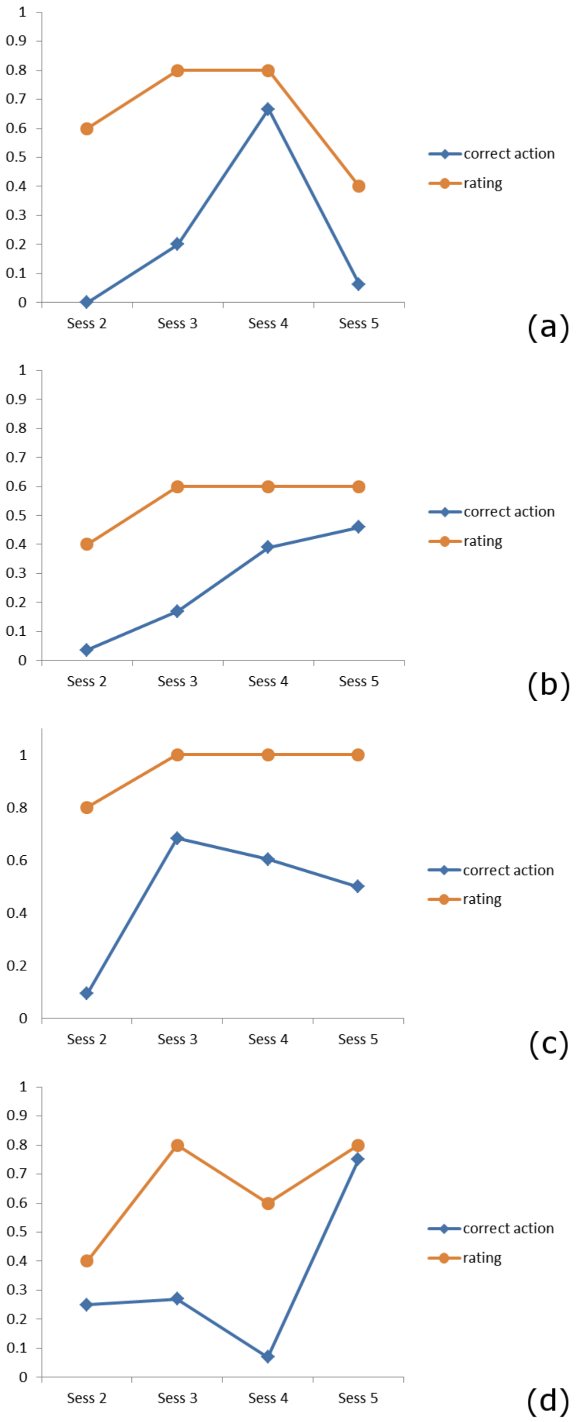


Figure 3: Relationship between action correctness and student rating

newer, more effective actions were presented to the student. This helps ensure that the student will always be prompted to select the most effective learning behavior. The relationship between the number of effective rules followed by the student and their learning effectiveness ratings indicate that the learning policy-based feedback will have a good chance of helping students learn more effectively.

The architecture we have designed still has some issues that need to be addressed. Our state representation did not contain information regarding students' metacognitive evaluations. Although we used emotions to approximate these evaluations, asking students to annotate them will be more accurate and create better policies. The reward values we used were based on students' self-evaluations and it would be interesting to see the difference when using learning gains instead (e.g., asking students to take a pretest and posttest). Combining both learning gains and self-evaluation to create the reward value may be a better measurement because it will consider both the student's preferred learning behavior and knowledge gained.

Our architecture also faces a common problem in RL called the exploration-exploitation problem. In order for the policy to be optimal, students need to try as much actions as possible. Due to the approach's reliance on the student's learning behavior, it cannot suggest actions outside of the current learning policy. This would require mechanisms for suggesting actions not in the learning policy such as using other students' learning policies or using expert knowledge.

Even though the approach can create policies that span across learning episodes, it has only been tested with learning episodes having the same goal. In the case of our data, students were either writing a conference paper or creating a power point presentation. It will be more useful if it could also be used across different learning goals. The current approach needs to be tested to see how well it fares in such a case and necessary modifications need to be applied accordingly.

The data we used was collected from adult learners and may be effective for them. However, according to Pressley *et. al.* [9], children have difficulty in verifying learning strategy utility even after practice. It is possible that additional feedback may be needed to fit this approach to younger learners.

Acknowledgements

This work was supported in part by the Management Expenses Grants for National Universities Corporations from the Ministry of Education, Culture, Sports, Science and Technology of Japan (MEXT) and JSPS KAKENHI Grant Number 23300059. We would also like to thank all the students who participated in our data collection.

7. REFERENCES

[1] S. Arai and K. Sycara. Effective learning approach for planning and scheduling in Multi-Agent domain. In *6th International Conference on Simulation of Adaptive Behavior*, pages 507–516, 2000.

[2] R. Azevedo, R. S. Landis, R. Feyzi-Behnagh, M. Duffy, G. Trevors, J. M. Harley, F. Bouchet, J. Burlison, M. Taub, N. Pacampara, M. Yeasin, A. K.

M. M. Rahman, M. I. Tanveer, and G. Hossain. The effectiveness of pedagogical agents' prompting and feedback in facilitating co-adapted learning with MetaTutor. In *Intelligent Tutoring Systems*, pages 212–221, 2012.

[3] C. S. Carver and M. F. Scheier. Origins and functions of positive and negative affect: A control-process view. *Psychological Review*, 97(1):19–35, 1990.

[4] S. D. Craig, A. C. Graesser, J. Sullins, and B. Gholson. Affect and learning: An exploratory look into the role of affect in learning with AutoTutor. *Journal of Educational Media*, 29(3):241–250, 2004.

[5] P. Ekman. Are there basic emotions? *Psychological Review*, 99(3):550–553, 1992.

[6] G. Gan, C. Ma, and J. Wu. *Data clustering: theory, algorithms, and applications*, volume 20. Society for Industrial and Applied Mathematics, 2007.

[7] P. S. Inventado, R. Legaspi, R. Cabredo, and M. Numao. Student learning behavior in an unsupervised learning environment. In *20th International Conference on Computers in Education*, pages 730–737, 2012.

[8] J. S. Kinnebrew, K. M. Loretz, and G. Biswas. A contextualized, differential sequence mining method to derive students' learning behavior patterns. *Journal of Educational Data Mining*, in press.

[9] M. Pressley, J. R. Levin, and E. S. Ghatala. Memory strategy monitoring in adults and children. *Journal of Verbal Learning and Verbal Behavior*, 23(2):270–288, 1984.

[10] J. Sabourin, L. R. Shores, B. W. Mott, and J. C. Lester. Predicting student self-regulation strategies in game-based learning environments. In *Intelligent Tutoring Systems*, pages 141–150, 2012.

[11] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. A Bradford Book, 1998.

[12] P. H. Winne. Self-regulated learning viewed from models of information processing. *Self-regulated learning and academic achievement: Theoretical perspectives*, 2:153–189, 2001.

[13] B. J. Zimmerman. Self-regulated learning and academic achievement: An overview. *Educational psychologist*, 25(1):3–17, 1990.

[14] B. J. Zimmerman. Becoming a Self-Regulated learner: An overview. *Theory Into Practice*, 41(2):64–70, 2002.

AIED 2013 Workshops Proceedings
Volume 9

The First Workshop on AI-supported Education for Computer Science (AIEDCS 2013)

Workshop Co-Chairs:

Nguyen-Thinh Le[1]
Kristy Elizabeth Boyer[2]
Beenish Chaudhry[3]
Barbara Di Eugenio[4]
Sharon I-Han Hsiao[5]
Leigh Ann Sudol-DeLyser[6]

[1]Clausthal University of Technology, Germany
[2]North Carolina State University, USA
[3]Indiana University Bloomington, USA
[4]University of Illinois Chicago, USA
[5]Columbia University, USA
[6]New York University, USA

https://sites.google.com/site/aiedcs2013/

Preface

The global economy increasingly depends upon Computer Science and Information Technology professionals to maintain and expand the infrastructure on which business, education, governments, and social networks rely. Demand is growing for a global workforce that is well versed and can easily adapt ever-increasing technology. For these reasons, there is increased recognition that computer science and informatics are becoming, and should become, part of a well-rounded education for every student. However, along with an increased number and diversity of students studying computing comes the need for more supported instruction and an expansion in pedagogical tools to be used with novices. The study of computer science often requires a large element of practice, often self-guided as homework or lab work. Practice as a significant component of the learning process calls for AI-supported tools to become an integral part of current course practices.

Designing and deploying AI techniques within computer science learning environments presents numerous challenges. First, computer science focuses largely on problem solving skills in a domain with an infinitely large problem space. Modeling possible problem solving strategies of experts and novices requires techniques that address many types of unique but correct solutions to problems. In addition, there is growing need to support affective and motivational aspects of computer science learning, to address widespread attrition of students from the discipline. AIED researchers are poised to make great strides in building intelligent, highly effective AI-supported learning environments and educational tools for computer science and information technology. Spurred by the growing need for intelligent learning environments that support computer science and information technology, this workshop will provide a timely opportunity to present emerging research results along these lines.

June, 2013
Nguyen-Thinh Le, Kristy Elizabeth Boyer, Beenish Chaudhry,
Barbara Di Eugenio, Sharon I-Han Hsiao, and Leigh Ann Sudol-DeLyser

Program Committee

Co-Chair: Nguyen-Thinh Le, *Clausthal University of Technology, Germany*
(nguyen-thinh.le@tu-clausthal.de)
Co-Chair: Kristy Elizabeth Boyer, *North Carolina State University, USA*
(keboyer@ncsu.edu)
Co-Chair: Beenish Chaudry, *Indiana University Bloomington, USA*
(bchaudry@indiana.edu)
Co-Chair: Barbara Di Eugenio, *University of Illinois Chicago, USA*
(bdieugen@uic.edu)
Co-Chair: Sharon I-Han Hsiao, *Columbia University*, *USA*
(ih2240@columbia.edu)
Co-Chair: Leigh Ann Sudol-DeLyser, *New York University, USA*
(leighannsudol@gmail.com)

James Lester, *North Carolina State University, USA*
Niels Pinkwart, *Clausthal University of Technology, Germany*
Peter Brusilovsky, *University of Pittsburgh, USA*
Michael Yudelson, *Carnegie Learning, USA*
Tomoko Kojiri, *Kansai University, Japan*
Fu-Yun Yu, *National Cheng Kung University, Taiwan*
Tsukasa Hirashima, *Hiroshima University, Japan*
Kazuhisa Seta, *Osaka Prefecture University, Japan*
Davide Fossati, *Carnegie Mellon University, Qatar*
Sergey Sosnovsky, *CeLTech, DFKI, Germany*
Tiffany Barnes, *North Carolina State University, USA*
Chad Lane, *USC Institute for Creative Technologies, USA*
Bruce McLaren, *Carnegie Mellon University, USA*
Pedro José Muñoz Merino, *Universidad Carlos III de Madrid, Spain*
Wei Jin, *University of West Georgia, USA*
John Stamper, *Carnegie Mellon University, USA*
Sajeesh Kumar, *University of Tennessee, USA*

Table of Contents

Sequential Patterns of Affective States of Novice Programmers

Nigel Bosch[1] and Sidney D'Mello[1,2]

Departments of Computer Science[1] and Psychology[2], University of Notre Dame
Notre Dame, IN 46556, USA
{pbosch1, sdmello}@nd.edu

Abstract. We explore the sequences of affective states that students experience during their first encounter with computer programming. We conducted a study where 29 students with no prior programming experience completed various programming exercises by entering, testing, and running code. Affect was measured using a retrospective affect judgment protocol in which participants annotated videos of their interaction immediately after the programming session. We examined sequences of affective states and found that the sequences Flow/Engagement ↔ Confusion and Confusion ↔ Frustration occurred more than expected by chance, which aligns with a theoretical model of affect during complex learning. The likelihoods of some of these frequent transitions varied with the availability of instructional scaffolds and correlated with performance outcomes in both expected but also surprising ways. We discuss the implications and potential applications of our findings for affect-sensitive computer programming education systems.

Keywords: affect, computer programming, computerized learning, sequences

1 Introduction

Given the unusually high attrition rate of computer science (CS) majors in the U.S. [1], efforts have been made to increase the supply of competent computer programmers through computerized education, rather than relying on traditional classroom education. Some research in this area focuses on the behaviors of computer programming students in order to provide more effective computerized tutoring and personalized feedback [2]. In fact, over 25 years ago researchers were exploring the possibility of exploiting artificial intelligence techniques to provide customized tutoring experiences for students in the LISP language [3]. This trend has continued, as evidenced by a number of intelligent tutoring systems (ITSs) that offer adaptive support in the domain of computer programming (e.g. [4–6]).

One somewhat neglected area in the field is the systematic monitoring of the affective states that arise over the course of learning computer programming and the impact of these states on retention and learning outcomes. The focus on affect is motivated by considerable research which has indicated that affect continually operates throughout a learning episode and different affective states differentially impact performance outcomes [7]. Some initial work has found that affective states, such as confusion and frustration, occur frequently during computer programming sessions [8, 9] and these states are correlated with student performance [10].

The realization of the important role of affect in learning has led some researchers to develop learning environments that adaptively respond to affective states in addition to cognitive states (see [11] for a review). Previous research has shown that affect sensitivity can make a measurable improvement on the performance of students in other domains such as computer literacy and conceptual physics [12, 13]. Applying this approach to computer programming education by identifying the affective states of students could yield similarly effective results, leading to more effective systems.

Before it will be possible for an affect-sensitive intelligent tutoring system to be successful in the computer programming domain, more research is needed to determine at a fine-grained level what affective states students experience and how affect interacts and arises from the students' behaviors. Previous work has collected affective data at a somewhat coarse-grained level in a variety of computer programming education contexts. [10] collected affect using two human observers, and were able to draw conclusions about what affective states led to improved performance on a computer programming exam. [14] induced affect in experienced programmers using video stimuli, and found that speed and performance on a coding and debugging test could be increased with high-arousal video clips.

In our previous work [15], we examined the affect of 29 novice programmers at 20-second intervals as they solved introductory exercises on fundamentals of computer programing. We found that flow/engagement, confusion, frustration, and boredom dominated the affect of novice programmers when they were not in a neutral state. We found that boredom and confusion were negatively correlated with performance, while the flow/engagement state positively predicted performance. This paper continues this line of research by exploring transitions between affective states.

Specifically, we test a theoretical model on affect dynamics that has been proposed for a range of complex learning tasks [16]. This theoretical model (Fig. 1) posits four affective states that are crucial to the learning process: flow/engagement, confusion, frustration, and boredom. The model predicts an important interplay between confusion and flow/engagement, whereby a learner in the state of flow/engagement may encounter an impasse and become confused. From the state of confusion, if an impasse is resolved the learner will return to the state of flow/engagement, having learned more deeply. This is in line with other research which has shown that confusion helps learning when impasses are resolved [17]. On the other hand, when the source of the confusion is never resolved, the learner will become frustrated, and eventually bored if the frustration persists.

Researchers have found some support for this theoretical model of affective dynamics in learning contexts such as learning computer literacy with AutoTutor [16], unsupervised academic research [18], and narrative learning environments [19]. We expect the theoretical model to apply to computer programming as well.

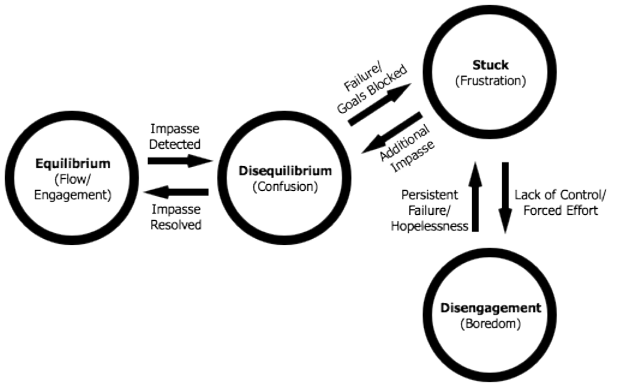


Fig. 1. Theoretical model of affect transitions.

We posit that encountering unfamiliar concepts, syntax and runtime errors, and other impasses can cause confusion in a computer programmer. When those impasses are resolved, the programmer will be better equipped to anticipate and handle such impasses in the future, having learned something. Alternatively, if the impasses persist, programmers may become frustrated and eventually disengage, entering a state of boredom in which it is difficult to learn.

To explore the applicability of this model to the domain of novice computer programming, this paper focuses on answering the following research questions: 1) what transitions occur frequently between affective states? 2) how are instructional scaffolds related to affect transitions? and 3) are affective transitions predictive of learning outcomes? These questions were investigated by analyzing affect data collected in a previous study [15] where 29 novice programmers learned the basics of computer programming over the course of a 40-minute learning session with a computerized environment, as described in more detail below.

2 Methods

Participants were 29 undergraduate students with no prior programming experience. They were asked to complete exercises in a computerized learning environment designed to teach programming fundamentals in the Python language. Participants solved exercises by entering, testing, and submitting code through a graphical user interface. Submissions were judged automatically by testing predetermined input and output values, whereupon participants received minimal feedback about the correctness of their submission. For correct submissions they would move on to the next exercise, but otherwise would be required to continue working on the same exercise.

The exercises in this study were designed in such a way that participants would likely encounter some unknown, potentially confusing concepts in each exercise. In this manner we elicited emotional reactions similar to real-world situations where computer programmers face problems with no predefined solutions and must experiment and explore to find correct solutions. Participants could use hints, which would gradually explain these impasses and allow participants to move on in order to prevent becoming permanently stuck on an exercise. However, participants were free to use or ignore hints as they pleased.

Exercises were divided into two main phases. In the first phase (scaffolding), participants had hints and other explanations available and worked on gradually more difficult exercises for 25 minutes. Performance in the scaffolding phase was determined by granting one point for each exercise solved and one point for each hint that was not used in the solved exercises. Following that was the second phase (fadeout), in which they had 5 minutes to work on a debugging exercise, and 10 minutes to work on another programming exercise with no hints. In this study we will not consider the debugging exercise because it was only 5 minutes long. Performance was determined by two human judges who examined each participant's code, determined the number of lines matching lines in the correct solution, and resolved their discrepancies.

Finally, we used a retrospective affect judgment protocol to assess student affect after they completed the 40-minute programming session [20]. Participants viewed video of their face and on-screen activity side by side, and were polled at various points to report the affective state they had felt most at the polling point. The temporal locations for polling were chosen to correspond with interactions and periods of no activity such that each participant had 100 points at which to rate their affect, with a minimum of 20 seconds between each point. Participants provided judgments on 13 emotions, including basic emotions (anger, disgust, fear, sadness, surprise, happiness), learning-centered emotions (anxiety, boredom, frustration, flow/engagement, curiosity, confusion/uncertainty) and neutral (no apparent feeling).The most frequent affective states, reported in [15], were flow/engagement (23%), confusion (22%), frustration (14%), and boredom (12%), a finding that offers some initial support for the theoretical model discussed in the Introduction.

3 Results and Discussion

We used a previously developed transition likelihood metric to compute the likelihood of the occurrence of each transition relative to chance [21].

$$L(Current \rightarrow Next) = \frac{\Pr(Next|Current) - \Pr(Next)}{1 - \Pr(Next)} \quad (1)$$

This likelihood metric determines the conditional probability of a particular affective state (*next*), given the current affective state. The probability is then normalized to account for the overall likelihood of the *next* state occurring. If the affective transition occurs as expected by chance, the numerator is 0 and so likelihood is as well. Thus we can discover affective state transitions that occurred more ($L > 0$) or less ($L < 0$) frequently than expected by chance alone.

Before computing L scores we removed transitions that occurred from one state to the same state. For example, a sequence of affective states such as *confusion, frustration, frustration, boredom* would be reduced to *confusion, frustration, boredom*. This was done because our focus in this paper is on the transitions between different affective states, rather than on the persistence of each affective state [16, 18]. Furthermore, although transition likelihoods between all 13 states (plus neutral) were computed, the present paper focuses on transitions between states specified in the theoretical model (boredom, confusion, flow/engagement, and frustration), which also happen to be the most frequent affective states.

What transitions occur frequently between affective states? We found the transitions that occurred significantly more than chance (*L* = 0) by computing affect transition likelihoods for individual participants and then comparing each likelihood to zero (chance) with a two-tailed one-sample *t-test.* Significant ($p < .05$) and marginally significant ($p < .10$) transitions are shown in Figure 2 and are aligned with the theoretical model on affect dynamics.

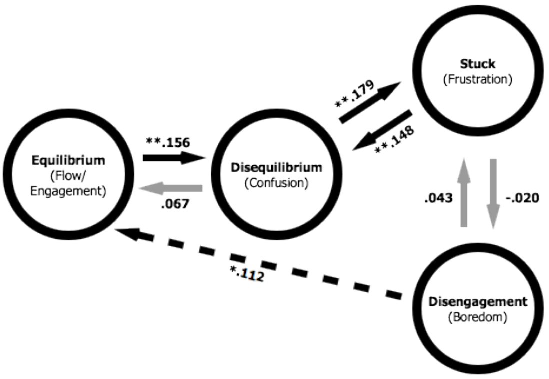


Fig 2. Frequently observed affective state transitions. Edge labels are mean likelihoods of affective state transitions. Grey arrows represent transitions that were predicted by the theoretical model but were not significant. The dashed arrow represents a transition that was marginally significant but not predicted. **p* < .10, ***p* < .05

Three of the predicted transitions, Flow/Engagement → Confusion, Confusion → Frustration, and Frustration → Confusion, were significant and matched the theoretical model. Confusion → Flow/Engagement was in the expected direction and approached significance (*p* = .108), while Boredom → Frustration was in the expected direction but not significant. The Frustration → Boredom transition was not in the expected direction and was also not significant. Hence, with the exception of the Frustration ↔ Boredom links, there was support for four out of the six transitions espoused by the theoretical model. This suggests that the components of the model related to the experience of successful (Flow/Engagement ↔ Confusion) and unsuccessful (Confusion ↔ Frustration links) resolution of impasses were confirmed. Therefore, the present data provide partial support for the model.

The Boredom → Flow/Engagement transition, which occurred at marginally significant levels (*p* = .091), was not predicted by the theoretical model. It is possible that the nature of our computerized learning environment encouraged this transition more than expected. This might be due to the fast-paced nature of the learning session, which included 18 exercises and an in-depth programming task in a short 40-minute session. Furthermore, participants had some control over the learning environment in that they could use bottom-out hints to move to the next exercise instead of being forced to wallow in their boredom. The previous study that tested this model used a learning environment (AutoTutor) that did not provide any control over the learning activity, which might explain the presence of Frustration → Boredom (dis-engaging from being stuck) and Boredom → Frustration (being frustrated due to forced effort) links in the earlier data [16].

How are instructional scaffolds related to affect transitions? To answer this question we looked at the differences between the scaffolding and fadeout phases of the study, as previously described. We discarded the first 5 minutes of the scaffolding phase to allow for a "warm-up" period during which participants were acclimating to the learning environment. We also discarded the 25 to 30 minutes portion, which was the debugging task in the fadeout phase. The debugging task was significantly different from the problem-solving nature of the coding portions, and so we excluded it from the current analysis to increase homogeneity. Differences between likelihoods of the five significant or marginally significant transitions from Figure 2 were investigated with paired samples *t*-test (see Table 1).

Table 1. Means and standard deviations (in parentheses) for common transitions in the scaffolding phase (5-25 minutes) and the coding portion of the fadeout phase (30-40 minutes).

Transition	**Scaffolding**	**Fadeout Coding**	**N**
Flow/Engagement → Confusion	**.115 (.308)	**.354 (.432)	20
Confusion → Flow/Engagement	.101(.241)	.029(.331)	27
Confusion → Frustration	.105 (.276)	.184 (.416)	27
Frustration → Confusion	.047 (.258)	.116 (.445)	21
Boredom → Flow/Engagement	.096 (.166)	.226 (.356)	14

**p* < .10, ***p* < .05

The likelihood of participants transitioning from flow/engagement to confusion was significantly higher in the fadeout phase compared to the scaffolding phase. This may be attributed to the fact that participants have hints and explanations in the scaffolding phase, so in the event of a confusing impasse, a hint may be helpful in resolving the impasse, thereby allowing participants to return to a state of flow/engagement. With no such hints, confused participants may become more frustrated in the fadeout phase, as evidenced by a trend in this direction. This finding is as expected from the theoretical model, which states that confusion can lead to frustration when goals are blocked and the student has limited coping potential (e.g. being unable to progress on an exercise in this case).

Although not significant, there also appears to be an increase in the Boredom → Flow/Engagement affect transition in the fadeout phase. It is possible that too much readily available assistance prevents students from re-engaging on their own.

Are affective transitions predictive of learning outcomes? To determine what affective state transitions were linked to performance on the programming task, we correlated the likelihood of affect transitions with the performance metrics described in the Methods. In previous work we found correlations between performance and the proportions of affective states experienced by students [15]. Hence, when examining the correlations between affect transitions and performance, partial correlations were used to control for the proportions of the affective states in the transitions.

Table 2 lists correlations between frequent transitions and performance. These include correlations between affect transitions in the scaffolding phase with performance in the scaffolding phase (Scaffolding column) and transitions in the fadeout phase with performance in the fadeout coding phase (Fadeout Coding 1). We also correlated transitions in the scaffolding phase with performance in the fadeout coding phase (Fadeout Coding 2). This allows us to examine if affect transitions experienced during scaffolded learning were related to future performance when learning scaffolds were removed. Due to the small sample size, in addition to discussing significant correlations, we also consider non-significant correlations approaching 0.2 or larger to be meaningful because these might be significant with a bigger sample. These correlations are bolded in the table.

Table 2. Correlations between affect transitions and performance.

Transition	**Scaffolding**	**Fadeout Coding 1**	**Fadeout Coding 2**
Flow/Engagement → Confusion	.046	-.094	-.098
Confusion → Flow/Engagement	__-.274__	__-.256__	*__-.365__
Confusion → Frustration	.114	**__.499__	**__.424__
Frustration → Confusion	*__-.368__	.051	__-.275__
Boredom → Flow/Engagement	-.034	.050	-.063

*$p < .10$, **$p < .05$

The correlations were illuminating in a number of respects. The Confusion → Flow/Engagement transition correlated *negatively* with performance. This is contrary to the theoretical model which would predict a positive correlation to the extent that confused learners return to a state of flow/engagement by resolving troublesome impasses with effortful problem solving. It is possible that students who frequently experienced this transition were doing so by taking advantages of hints as opposed to resolving impasses on their own. This would explain the negative correlation between Confusion → Flow/Engagement and performance.

To investigate this possibility we correlated hint usage in the scaffolding phase with the Confusion → Flow/Engagement transition, controlling for the proportion of confusion and flow/engagement. The number of hints used in the scaffolding phase correlated positively, though not significantly, with the Confusion → Flow/Engagement transition in the scaffolding phase (r = .297) and the fadeout coding phase (r = .282). Additionally, hint usage correlated negatively with score in the scaffolding phase (r = -.202) and the fadeout coding phase (r = -.506). This indicates that students using hints tended to experience the Confusion → Flow/Engagement transition more (as expected) but this hindered rather than helped learning because students were not investing the cognitive effort to resolve impasses on their own.

Similarly, the correlation between Confusion → Frustration and performance is inconsistent with the theoretical model, which would predict a negative relationship between these variables. This unexpected correlation could also be explained on the basis of hint usage. Specifically, the number of hints used in the scaffolding phase correlated negatively, though not significantly, with the Confusion → Frustration transition in the scaffolding phase (*r* = -.258) and the fadeout coding phase (*r* = -.171). This finding suggests that although hints can alleviate the Confusion → Frustration transition, learning improved when students are able to resolve impasses on their own, which is consistent with the theoretical model.

Finally, the correlation between Frustration → Confusion was in the expected direction. The Frustration → Confusion transition occurs when a student experience additional impasses while in the state of frustration. This transition is reflective of hopeless confusion, which is expected to be negatively correlated with performance, as revealed in the data.

4 General Discussion

Previous research has shown that some affective states are conducive to learning in the context of computer programming education while others hinder learning. Flow/engagement is correlated with higher performance, while confusion and boredom are correlated with poorer performance [10, 15]. Transitions between affective states are thus important because they provide insight into how students enter into an affective state. Affect-sensitive ITSs for computer programming may be able to use this information to better predict affect, intervening when appropriate to encourage the flow/engagement state and minimize the incidence of boredom and frustration.

We found that the presence or absence of instructional scaffolds were related the affect transitions experienced by students, especially the Flow/Engagement → Confusion transition. Our findings show that this transition is related to the presence of hints, a strategy which might be useful in future affect-sensitive ITS design for computer programming students. Similarly, we found that instructional scaffolds were related to the Boredom → Flow/Engagement transition, which is not part of the theoretical model. Future work on ITS design might also need to take into account this effect and moderate the availability of scaffolds to promote this affect transition.

The affect transitions that we found partially follow the predictions of the theoretical model. Impasses commonly arise in computer programming, particularly for novices, when they encounter learning situations with which they are unfamiliar. New programming language keywords, concepts, and error messages present students with impasses that must be resolved before the student will be able to continue. Unresolved impasses can lead to frustration and eventually boredom. The alignment between the theoretical model and the present data demonstrates the model's applicability and predictive power in the context of learning computer programming.

That being said, not all of the affect transitions we found matched predictions of the theoretical model. This includes lack of data to support the predicted Frustration → Boredom and Boredom → Frustration transitions and the presence of an unexpected Boredom → Flow/Engagement transition. Limitations with this study are likely responsible for some of these mismatches. The sample size was small, so it is possible that increased participation in the study might confirm some of these expected transitions. In particular, the Boredom → Frustration transition was in the predicted direction but not significant in our current sample. Additionally, we exclusively focused on affect, but ignored the intermediate events that trigger particular affective states (e.g., system feedback, hint requests, etc.). We plan to further explore our data by incorporating these interaction events as possible triggers for the observed transitions between affective states. This will allow us to more deeply understand why some of the predicted transitions did not occur (e.g., Frustration → Boredom) and some unexpected transitions did (e.g., Boredom → Flow/Engagement).

It is also possible that some aspects of the model might need refinement. In particular there appears to be an important relationship between Confusion → Frustration transitions, Confusion → Flow/Engagement transitions, performance, and hint usage. While hints may allow students to move past impasses and re-enter a state of flow/engagement, they may lead to an illusion of impasse resolution, which is not useful for learning. Conversely, resolving impasses without relying on external hints might lead a confused learner to momentarily experience frustration, but ultimately improve learning. Future work that increases sample size and specificity of the data (i.e., simultaneously modeling dynamics of affect and interaction events) will allow us to further explore the interaction of hints with the theoretical model, and is expected to yield a deeper understanding of affect dynamics during complex learning.

Acknowledgment. This research was supported by the National Science Foundation (NSF) (ITR 0325428, HCC 0834847, DRL 1235958). Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NSF.

References

1. Haungs, M., Clark, C., Clements, J., Janzen, D.: Improving first-year success and retention through interest-based CS0 courses. Proceedings of the 43rd ACM technical symposium on Computer Science Education. pp. 589–594. ACM, New York, NY, USA (2012).
2. Fossati, D., Di Eugenio, B., Brown, C.W., Ohlsson, S., Cosejo, D.G., Chen, L.: Supporting Computer Science Curriculum: Exploring and Learning Linked Lists with iList. IEEE Transactions on Learning Technologies. 2, 107–120 (2009).
3. Anderson, J.R., Skwarecki, E.: The automated tutoring of introductory computer programming. Communications of the ACM. 29, 842–849 (1986).
4. Brusilovsky, P., Sosnovsky, S., Yudelson, M.V., Lee, D.H., Zadorozhny, V., Zhou, X.: Learning SQL programming with interactive tools: From integration to personalization. ACM Transactions on Computing Education. 9, 19:1–19:15 (2010).
5. Cheung, R., Wan, C., Cheng, C.: An ontology-based framework for personalized adaptive learning. Advances in Web-Based Learning–ICWL 2010. pp. 52–61. Springer, Berlin Heidelberg (2010).
6. Hsiao, I.-H., Sosnovsky, S., Brusilovsky, P.: Guiding students to the right questions: adaptive navigation support in an E-Learning system for Java programming. Journal of Computer Assisted Learning. 26, 270–283 (2010).
7. Pekrun, R.: The impact of emotions on learning and achievement: Towards a theory of cognitive/motivational mediators. Applied Psychology. 41, 359–376 (1992).

8. Grafsgaard, J.F., Fulton, R.M., Boyer, K.E., Wiebe, E.N., Lester, J.C.: Multimodal analysis of the implicit affective channel in computer-mediated textual communication. Proceedings of the 14th ACM international conference on Multimodal interaction. pp. 145–152. ACM, New York, NY, USA (2012).
9. Lee, D.M.C., Rodrigo, M.M.T., Baker, R.S.J. d, Sugay, J.O., Coronel, A.: Exploring the relationship between novice programmer confusion and achievement. In: D'Mello, S., Graesser, A., Schuller, B., and Martin, J.C. (eds.) Affective Computing and Intelligent Interaction. pp. 175–184. Springer, Berlin Heidelberg (2011).
10. Rodrigo, M.M.T., Baker, R.S.J. d, Jadud, M.C., Amarra, A.C.M., Dy, T., Espejo-Lahoz, M.B.V., Lim, S.A.L., Pascua, S.A.M.S., Sugay, J.O., Tabanao, E.S.: Affective and behavioral predictors of novice programmer achievement. SIGCSE Bulletin. 41, 156–160 (2009).
11. D'Mello, S., Graesser, A.: Feeling, thinking, and computing with affect-aware learning technologies. In: Calvo, R.A., D'Mello, S., Gratch, J., and Kappas, A. (eds.) Handbook of Affective Computing. Oxford University Press (in press).
12. D'Mello, S., Lehman, B., Sullins, J., Daigle, R., Combs, R., Vogt, K., Perkins, L., Graesser, A.: A time for emoting: When affect-sensitivity is and isn't effective at promoting deep learning. In: Aleven, V., Kay, J., and Mostow, J. (eds.) Intelligent Tutoring Systems. pp. 245–254. Springer, Berlin Heidelberg (2010).
13. Forbes-Riley, K., Litman, D.: Benefits and challenges of real-time uncertainty detection and adaptation in a spoken dialogue computer tutor. Speech Communication. 53, 1115–1136 (2011).
14. Khan, I.A., Hierons, R.M., Brinkman, W.P.: Mood independent programming. Proceedings of the 14th European Conference on Cognitive Ergonomics: Invent! Explore! pp. 28–31. ACM, New York, NY, USA (2007).
15. Bosch, N., D'Mello, S., Mills, C.: What Emotions Do Novices Experience During their First Computer Programming Learning Session? Proceedings of the 16th International Conference on Artificial Intelligence in Education (AIED 2013) (in press).
16. D'Mello, S., Graesser, A.: Dynamics of affective states during complex learning. Learning and Instruction. 22, 145–157 (2012).
17. D'Mello, S., Lehman, B., Pekrun, R., Graesser, A.: Confusion can be beneficial for learning. Learning and Instruction. (in press).
18. Inventado, P.S., Legaspi, R., Cabredo, R., Numao, M.: Student learning behavior in an unsupervised learning environment. Proceedings of the 20th International Conference on Computers in Education. pp. 730–737. National Institute of Education, Singapore (2012).
19. McQuiggan, S.W., Robison, J.L., Lester, J.C.: Affective transitions in narrative-centered learning environments. In: Woolf, B.P., Aïmeur, E., Nkambou, R., and Lajoie, S. (eds.) Intelligent Tutoring Systems. pp. 490–499. Springer, Berlin Heidelberg (2008).
20. Rosenberg, E.L., Ekman, P.: Coherence between expressive and experiential systems in emotion. Cognition & Emotion. 8, 201–229 (1994).
21. D'Mello, S., Taylor, R.S., Graesser, A.: Monitoring affective trajectories during complex learning. Proceedings of the 29th annual meeting of the cognitive science society. pp. 203–208. Cognitive Science Society, Austin, TX (2007).

Towards Deeper Understanding of Syntactic Concepts in Programming

Sebastian Gross, Sven Strickroth, Niels Pinkwart, and Nguyen-Thinh Le

Clausthal University of Technology, Department of Informatics
Julius-Albert-Str. 4, 38678 Clausthal-Zellerfeld, Germany
{sebastian.gross,sven.strickroth}@tu-clausthal.de
{niels.pinkwart,nguyen-thinh.le}@tu-clausthal.de

Abstract. Syntactic mistakes and misconceptions in programming can have a negative impact on students' learning gains, and thus require particular attention in order to help students learn programming. In this paper, we propose embedding a discourse on syntactic issues and student's misconceptions into a dialogue between a student and an intelligent tutor. Based on compiler (error) messages, the approach aims to determine the cause for the error a student made (carelessness, misconception, or lack of knowledge) by requesting explanations for the violated syntactic construct. Depending on that cause, the proposed system adapts dialogue behaviours to student's needs by asking her to reflect on her knowledge in a self-explanation process, providing error-specific explanations, and enabling her to fix the error herself. This approach is designed to encourage students to develop a deeper understanding of syntactic concepts in programming.

Keywords: intelligent tutoring systems, programming, dialogue-based tutoring

1 Introduction

Programming is a useful skill and is related to several fields of study as economy, science, or information technology. Thus, teaching basics of programming is part of many curricula in universities and higher education. Programming is often taught bottom-up: First, syntactic aspects and low-level concepts are presented to students (e. g. variable declarations, IF, WHILE constructs, . . . in the object-oriented programming paradigm). Then, iteratively higher-level concepts are taught (e. g. methods, recursion, usage of libraries, . . .). Learning a programming language, however, cannot be approached theoretically only. It requires a lot of practice for correct understanding of abstract concepts (technical expertise) as well as logical and algorithmic thinking in order to map real-world problems to program code. Studies [8, 17] and our own teaching experiences have shown that studying programming is not an easy task and many students already experience (serious) difficulties with the basics: writing syntactically correct programs which can be processed by a compiler.

Source code is the basis for all programs, since without it algorithms cannot be executed and tested. Here, testing does not only mean testing done by students themselves. Often tutorial and/or submission systems [7, 18] are used by lecture advisors in order to optimize their workflow and to provide students some further testing opportunities. These tests often focus on the algorithms, check program outputs given a specific input and require runnable source code.

Creating correct source code requires good knowledge and strict observance of the syntax and basic constructs of the programming language. Yet, students often use an integrated development environment (IDE) from the very beginning. Here, code templates and also possible solutions for syntactic errors are offered. Based on our experience over several years of teaching a course on "Foundations of programming" in which Java is introduced and used as a main programming language, we suppose that these features (code templates provided by an IDE) possibly hinder learning and deeper understanding: Novice programmers seem to use these features and suggestions (which are actually addressed to people who already internalized the main syntactic and semantic concepts of programming) blindly. As a result, students are often not able to write programs on their own (e. g. on paper) and do not understand the cause of errors.

In this paper, we propose a new tutoring approach which initiates a dialogue-based discourse between a student and an intelligent tutor in case of a syntactic error. The intelligent tutor aims at detecting a possible lack of knowledge or an existing misconception as well as suggesting further readings and correcting the misconception, respectively. The remainder of this paper is organized as follows: First, in Section 2, we give an overview of the state of the art of intelligent learning systems in programming. In Section 3, we then describe our approach in more detail, illustrate an exemplary discourse, and characterize possible approaches for an implementation. Finally, we discuss our approach in Section 4, draw a conclusion and point out future work in Section 5.

2 Intelligent Learning Systems in Programming

In recent years, Intelligent Tutoring Systems (ITSs) have found their way increasingly into classrooms, university courses, military training and professional education, and have been successfully applied to help humans learn in various domains such as algebra [10], intercultural competence [16], or astronaut training [1]. Constraint-based and cognitive tutor systems are the most established concepts to build ITSs, and have shown to have a positive impact on learning [14]. In the domain of programming, several approaches have been successfully applied to intelligently support teaching of programming skills using artificial intelligence (AI) techniques. In previous work [12], we reviewed AI-supported tutoring approaches for programming: example-based, simulation-based, collaboration-based, dialogue-based, program analysis-based, and feedback-based approaches.

Several approaches for building ITSs in the domain of programming are based on information provided by compilers. The Expresso tool [6] supports students in identifying and correcting Java programming errors by interpreting Java compiler error messages and providing feedback to students based on these messages. JECA is a Java error correcting algorithm which can be used in Intelligent Tutoring Systems in order to help students find and correct mistakes [19]. The corresponding system prompted learner whether or not the system shall automatically correct found errors. Coull and colleagues [3] suggested error solutions to learners based on compiler messages by parsing these messages and comparing them to a database. These approaches aim to support learners in finding and correcting syntactic errors without explicitly explaining these issues, and, thus, did not ensure that a learner internalizes the underlying concept. HelpMeOut [5], however, is a recommender system based on compiler messages and runtime exceptions which formulated queries to a database containing error-specific information in order to recommend explanations for students' mistakes. The underlying database could be extended by users' input generated via peer interactions. This approach did not allow a discourse in order to determine student's knowledge or to correct possible misconceptions in student's application of knowledge, but provides solutions to students without encouraging students' learning. In our approach, we propose a dialogue-based discourse between a student and a tutor which aims at identifying the cause of the syntactic error, and at ensuring that the student gains a deeper understanding of the underlying syntactic concept she violated.

3 Solution Proposal

Programmers need to master syntactic and semantic rules of a programming language. Using integrated development environments such as Eclipse or Netbeans supports experienced programmers in finding and correcting careless mistakes and typos, and thus help them to efficiently focus on semantic issues. Novice programmers, however, who are still learning a programming language and, thus, are probably not entirely familiar with the syntactic concepts might be overwhelmed by messages provided by compilers. Interpreting error messages and correcting mistakes based on these messages can be a frustrating part of programming for those learners. IDEs, indeed, help them finding and correcting an error, but also impede learner's learning if learners follow IDEs' suggestion without reflecting on these hints and understanding why an error occurred.

How well programmers are able to find and correct syntactic mistakes strongly depends on the quality of messages and hints provided by compilers or IDEs [2, 13, 15]. Following previous work in the field of intelligent supporting systems for programming, we propose to provide guidance to novice programmers based on compiler (error) messages in order to help them master syntactic issues of programming languages. Instead of enriching compiler messages, we aim to determine student's knowledge about a specific violated syntactic construct. Depending on a student's level of knowledge, we propose to adapt the system's learning support to student's individual needs. For this, we distinguish three causes for syntactic errors:

E1 Errors caused by carelessness,
E2 Errors caused by lack of knowledge,
E3 Errors caused by misconceptions.

In order to determine which one of the three causes applies to a specific error, we propose to initiate a discourse between the learner and an intelligent tutor (shown in Figure 1). Information provided by a compiler can be used to identify an erroneous part and the syntactic concept the student violated in order to lead the discourse to corresponding syntactic aspects. Embedded in dialogues and backed up by a knowledge database, the tutor first aims to determine whether or not the student is able to explain the underlying concept of the violated statement or syntactic expression. Our approach requires a knowledge base of the most typical errors of students. For this purpose, we used data collected in the submission system GATE [18]. We used GATE in our introductory Java teaching courses since 2009. This system supports the whole workflow from task creation, file submission, (limited) automated feedback for students to grading. We analyzed and categorized 435 compiler outputs of failed Java code compilations of student solutions: The ten most common syntax errors according to the compiler outputs (covering 70 % of all errors) are missing or superfluous braces (56 cases), usage of missing classes (e. g. based on an incomplete upload; 45), mismatching class names (according to the file name; 37), usage of undeclared variables (35), problems with if-constructs (23), usage of incompatible types (21), method definitions within other methods (primarily within the main method; 19), usage of undeclared methods (18), missing return statements in methods (14), and problems with SWITCH statements (12).

Just as experienced programmers also novice programmers make mistakes which are caused by carelessness (**E1**, e. g. a typo). In this case students are able to correctly and completely explain the concepts. The tutor then confirms the student's correct explanation, and students are able to fix the error without any further help. Errors caused by lacks of knowledge or misconception in the application of the knowledge, however, require special attention. This is the case if the student is not able to correctly and/or completely explain the underlying concept of a statement or syntactic expression which was violated. Then the tutor is not able to recognize student's explanation and distinguishes whether

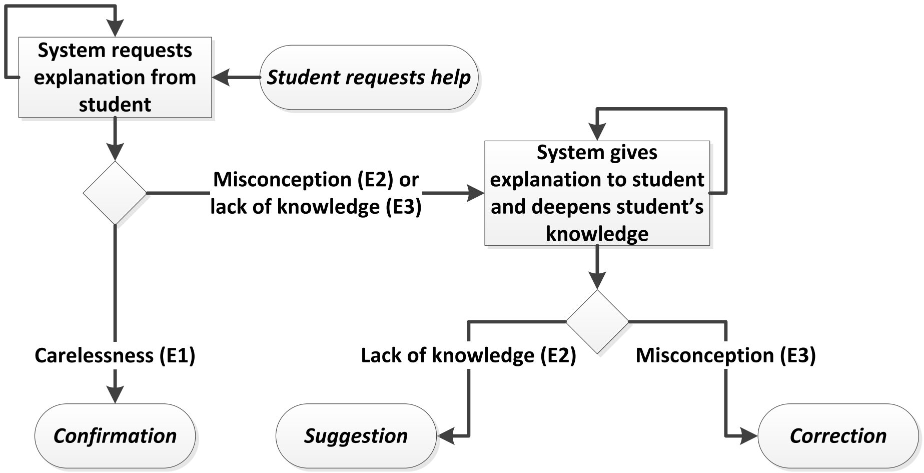

System requests explanation from student
Student requests help
Misconception (E2) or lack of knowledge (E3)
System gives explanation to student and deepens student's knowledge
Carelessness (E1)
Lack of knowledge (E2)
Misconception (E3)
Confirmation
Suggestion
Correction

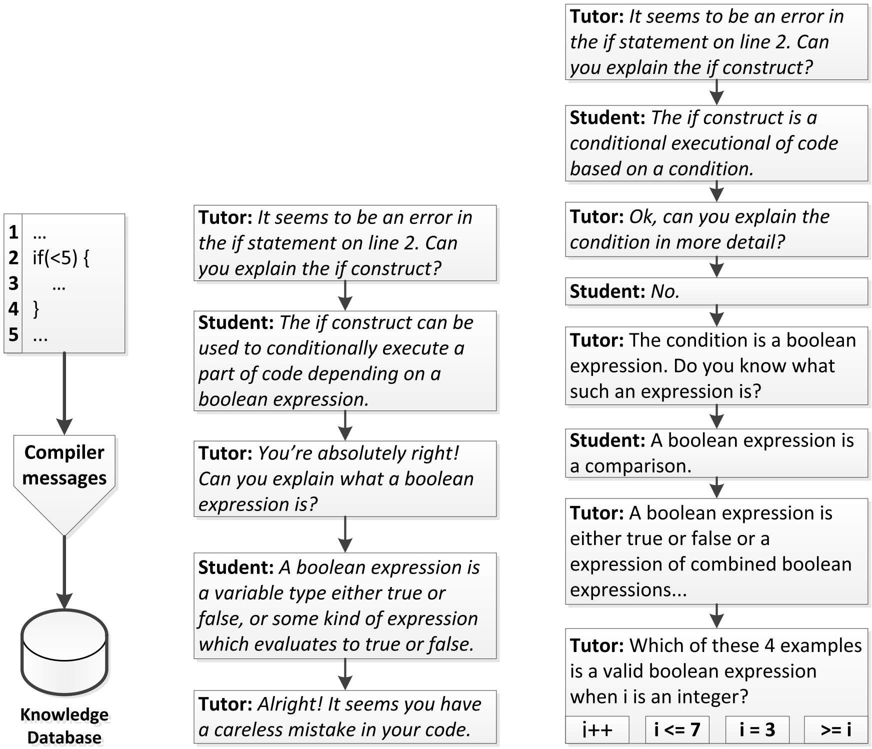
Fig. 1: Dialogue-based identification of cause for syntactic error

a lack of knowledge or a misconception caused the error by requesting further explanations from students. If a lack of knowledge is detected (**E2**), the tutor then suggests how to correct the error or points to the part of a (video) lecture explaining the violated concept. In the other case, if a misconception is detected (**E3**), the tutor changes its role in the discourse in order to revise the student's wrong and/or incomplete explanation. In this error-specific dialogue, the tutor then tries to explain the underlying concept the student violated. Therefore, the tutor could then provide step-by-step explanations using the knowledge base. To evaluate student understanding of single steps of explanations, the tutor could ask the student to confirm whether or not she understood the explanation, to ask her to complete/correct incomplete/erroneous examples covering the underlying syntactic concept, or to assess student's knowledge in question and answer manner.

In summary, we propose a dialogue-based intelligent tutor which initially interprets compiler (error) messages in order to identify the syntactic concept the student violated. Based on the compiler information, the tutor initiates a discourse with the student where it determines the cause of the error (**E1**, **E2** or **E3**). In a deeper examination of student's knowledge, the tutor uses a knowledge base in order to impart and deepen the concept which the syntactic error corresponds to. The tutor uses a computational model that is capable of automatically evaluating student's responses on tutor's questions. The goal is to correct misconceptions or to suggest further readings in order to fill lacks of knowledge and enable students to fix their mistakes in their own this way. In Section 3.2, we explain how such a model can be implemented.

3.1 Exemplary Dialogue-Based Discourse

In the above, we introduced typical syntactic errors that were made by students who attended a course on "Foundations of programming". The dataset contained students' exercise submissions of one of our introductory Java courses. To illustrate our approach (described in Section 3), we discuss a dialogue-based discourse exemplary for one of those typical errors (see Figure 2). A typical error that often occurred in students' submissions was that the implementation of a condition statement (IF construct) did not match the underlying syntactic concept. In the first dialogue (shown in Figure 2b), the tutor asks the student to explain the IF construct and, because it is part of an if-statement, what a boolean expression is. Here, the student is able to explain both concepts, and thus the mistake seems to have been caused by carelessness and the tutor confirms the student's explanations. In the second dialogue (shown in Figure 2c), the student gives an incomplete explanation on tutor's request. The tutor, consequently, asks the student to explain the condition in more detail which the student is not able to do. At that point, the tutor switches from requesting to providing explanations, and aims at deepening student's knowledge. Finally, the tutor aims at evaluating whether the student understood its explanations by asking a multiple-choice-question. Depending on the student's answer, the tutor can then assess whether the error was caused by a misconception or lack of knowledge. In the one case, the student is able to correctly respond to tutor's



```
1 ...
2 if(<5) {
3     ...
4 }
5 ...
```

Tutor: *It seems to be an error in the if statement on line 2. Can you explain the if construct?*

Student: *The if construct can be used to conditionally execute a part of code depending on a boolean expression.*

Tutor: *You're absolutely right! Can you explain what a boolean expression is?*

Student: *A boolean expression is a variable type either true or false, or some kind of expression which evaluates to true or false.*

Tutor: *Alright! It seems you have a careless mistake in your code.*

(b) Dialogue 1

Tutor: *It seems to be an error in the if statement on line 2. Can you explain the if construct?*

Student: *The if construct is a conditional executional of code based on a condition.*

Tutor: *Ok, can you explain the condition in more detail?*

Student: *No.*

Tutor: The condition is a boolean expression. Do you know what such an expression is?

Student: A boolean expression is a comparison.

Tutor: A boolean expression is either true or false or a expression of combined boolean expressions...

Tutor: Which of these 4 examples is a valid boolean expression when i is an integer?

(c) Dialogue 2

Fig. 2: Dialogue-based discourse between student and intelligent tutor

question which indicates a misconception that could be corrected during the discourse. In the other case, the student is not able to correctly respond to the tutor's question which indicates lack of knowledge. Here, the tutor might suggest the student to repeat appropriate lecture(s)/exercise(s) in order to acquire the necessary knowledge.

3.2 Technical Implementation

In the dialogue-based approach proposed in this paper we need to distinguish two types of student's answers. The first one consists of explanations about a concept upon request of the system, and the second one includes short answers on error-specific examples and questions.

In order to understand a student's explanation on a programming concept we either provide her options to be chosen or allow her to express the explanation in a free form. In the first case, the system can understand the student's explanation by associating each template with a classifier of the error type. For example, in order to determine whether the student has made an error in the IF condition statement by carelessness, by misconception or lack of knowledge, we can ask the student to explain this concept and provide her with three possible answers: 1) *The IF construct can be used to conditionally execute a part of code depending on a boolean expression.*, 2) *The IF construct can be used to express factual implications, or hypothetical situations and their consequences.*, 3) *I have no idea.* Obviously, the first answer is correct and the second answer is a misconception because students might refer the IF construct of a programming language (e. g., Java) to the IF used in conditional sentences in the English language. The third option indicates that the student has lack of the condition concept. This approach seems to be easy to implement, but requires a list of typical misconceptions of students. If we allow the student to express an explanation in a free form, the challenge is to understand possible multi-sentential explanations. In order to deal with this problem Jordan and colleagues [9] suggested to process explanations through two steps: 1) single sentence analysis, which outputs a first-order predicate logic representation, and 2) then assessing the correctness and completeness of these representations with respect to nodes in correct and buggy chains of reasoning.

In order to understand short answers on error-specific examples and questions, we can apply the form-filling approach for initiating dialogues. That is, for each question/example, correct answers can be anticipated and authored in the dialogue system. This approach is commonly used in several tutoring systems, e. g., the dialogue-based EER-Tutor [20], PROPL [11], AUTOTUTOR [4]. In addition to the form-filling approach, the Latent Semantic Analysis technique can also be deployed to check the correctness in the natural language student's answer by determining which concepts are present in a student's utterance (e. g., AUTOTUTOR).

4 Discussion

Our approach relies on the compiler's output. So, ambiguity of compiler messages is a crucial issue (also for students). The standard Java compiler works by following a greedy policy which causes that errors are reported for the first position in the source code where the compiler recognized a mismatch despite the fact that the cause of the error might lie somewhere else. There are also different parsers that use other policies and are capable of providing more specific feedback (e. g. the parser of the Eclipse IDE). Taking the code fragment "int i : 5;", e.g., the standard Java compiler outputs that it expects a ";" instead of the colon. The Eclipse compiler, however, outputs, that the colon is wrong and suggests that the programmer might have wanted to use the equal character "=". This difference in the compilers becomes even more manifest for lines where an opening brace is included. If there is an error in this line before the brace, the whole line is ignored by the standard Java compiler and a superfluous closing brace is reported at the end of the source code. Here, using a better parser (or even a custom parser) could improve error recognition regarding the position of the error and the syntactic principles violated by the programmer. Additional and more detailed information can help to cover more syntactic issues and to apply a more sophisticated discourse between learners and a dialogue-based tutor.

Generally, it is sufficient for our approach that a compiler reports the correct line and the affected basic structure of an error (e. g. If-statement), since our approach does not aim for directly solving the error, but supporting the students to fix the mistake on their own. This, however, requires a good knowledge base of the basic structures about a programming language.

5 Conclusion and Future Work

In this paper, we proposed a dialogue-based approach interpreting compiler (error) messages in order to determine syntactic errors students made, and thus to adapt the behaviour of the intelligent tutor to the individual needs of students depending on three causes of errors (carelessness, lack of knowledge, or misconception). Our proposed system initiates a dialogue asking for explanations of the violated syntactic construct and determines which cause applies for the affected violated construct. Then the proposed approach adapts dialogue behaviours to student's needs confirming correct knowledge or providing error-specific explanations. We argued that this method works better than just presenting error messages or suggestions for fixing an error, because it encourages students to reflect on their knowledge in a self-explanation process and finally enables them to fix the errors themselves.

In future, we plan to implement our approach and test it with students in an introductory programming course. Initially, we will apply self-explanation in human-tutored exercises in order to gather dialogues which can be used to build a model for our approach.

References

[1] K. Belghith, R. Nkambou, F. Kabanza, and L. Hartman. An intelligent simulator for telerobotics training. *IEEE Transactions on Learning Technologies*, 5(1):11–19, 2012.

[2] B. Boulay and I. Matthew. Fatal error in pass zero: How not to confuse novices. In G. Veer, M. Tauber, T. Green, and P. Gorny, editors, *Readings on Cognitive Ergonomics Mind and Computers*, volume 178 of *Lecture Notes in Computer Science*, pages 132–141. Springer Berlin Heidelberg, 1984.

[3] N. Coull, I. Duncan, J. Archibald, and G. Lund. Helping Novice Programmers Interpret Compiler Error Messages. In *Proceedings of the 4th Annual LTSN-ICS Conference*, pages 26–28. National University of Ireland, Galway, Aug. 2003.

[4] A. Graesser, N. K. Person, and D. Harter. Teaching Tactics and Dialog in AutoTutor. *International Journal of Artificial Intelligence in Education*, 12:257–279, 2001.

[5] B. Hartmann, D. MacDougall, J. Brandt, and S. R. Klemmer. What would other programmers do: suggesting solutions to error messages. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1019–1028, New York, NY, USA, 2010. ACM.

[6] M. Hristova, A. Misra, M. Rutter, and R. Mercuri. Identifying and correcting java programming errors for introductory computer science students. In *Proceedings of the 34th SIGCSE technical symposium on Computer science education*, SIGCSE '03, pages 153–156, New York, NY, USA, 2003. ACM.

[7] P. Ihantola, T. Ahoniemi, V. Karavirta, and O. Seppälä. Review of recent systems for automatic assessment of programming assignments. In *Proceedings of the 10th Koli Calling International Conference on Computing Education Research*, Koli Calling '10, pages 86–93, New York, NY, USA, 2010. ACM.

[8] T. Jenkins. A participative approach to teaching programming. In *Proceedings of the 6th annual conference on the teaching of computing and the 3rd annual conference on Integrating technology into computer science education: Changing the delivery of computer science education*, ITiCSE '98, pages 125–129, New York, NY, USA, 1998. ACM.

[9] P. W. Jordan, M. Makatchev, U. Pappuswamy, K. VanLehn, and P. L. Albacete. A natural language tutorial dialogue system for physics. In G. Sutcliffe and R. Goebel, editors, *FLAIRS Conference*, pages 521–526. AAAI Press, 2006.

[10] K. R. Koedinger, J. R. Anderson, W. H. Hadley, and M. A. Mark. Intelligent tutoring goes to school in the big city. *International Journal of AI in Education*, 8:30–43, 1997.

[11] H. C. Lane and K. VanLehn. A dialogue-based tutoring system for beginning programming. In V. Barr and Z. Markov, editors, *FLAIRS Conference*, pages 449–454. AAAI Press, 2004.

[12] N. T. Le, S. Strickroth, S. Gross, and N. Pinkwart. A review of AI-supported tutoring approaches for learning programming. In *Accepted for the International Conference on Computer Science, Applied Mathematics and Applications 2013, Warsaw, Poland*. Springer Verlag, 2013.

[13] G. Marceau, K. Fisler, and S. Krishnamurthi. Measuring the effectiveness of error messages designed for novice programmers. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, SIGCSE '11, pages 499–504, New York, NY, USA, 2011. ACM.

[14] A. Mitrovic, K. Koedinger, and B. Martin. A comparative analysis of cognitive tutoring and constraint-based modeling. In P. Brusilovsky, A. Corbett, and F. Rosis, editors, *User Modeling 2003*, volume 2702 of *Lecture Notes in Computer Science*, pages 313–322. Springer Berlin Heidelberg, 2003.

[15] M.-H. Nienaltowski, M. Pedroni, and B. Meyer. Compiler error messages: what can help novices? In *Proceedings of the 39th SIGCSE technical symposium on Computer science education*, SIGCSE '08, pages 168–172, New York, NY, USA, 2008. ACM.

[16] A. Ogan, V. Aleven, and C. Jones. Advancing development of intercultural competence through supporting predictions in narrative video. *International Journal of AI in Education*, 19(3):267–288, Aug. 2009.

[17] A. Robins, J. Rountree, and N. Rountree. Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2):137–172, 2003.

[18] S. Strickroth, H. Olivier, and N. Pinkwart. Das GATE-System: Qualitätssteigerung durch Selbsttests für Studenten bei der Onlineabgabe von Übungsaufgaben? In *DeLFI 2011: Die 9. e-Learning Fachtagung Informatik*, number P-188 in GI Lecture Notes in Informatics, pages 115 – 126. GI, 2011.

[19] E. R. Sykes and F. Franek. Presenting jeca: a java error correcting algorithm for the java intelligent tutoring system. In *Proceedings of the IASTED Conference on Advances in Computer science*, 2004.

[20] A. Weerasinghe, A. Mitrovic, and B. Martin. Towards individualized dialogue support for ill-defined domains. *International Journal of AI in Education*, 19(4):357–379, Dec. 2009.

An Intelligent Tutoring System for Teaching FOL Equivalence

Foteini Grivokostopoulou, Isidoros Perikos, Ioannis Hatzilygeroudis

University of Patras, Department of Computer Engineering &Informatics, 26500, Hellas (Greece)

{grivokwst,perikos,ihatz @ceid.upatras.gr}

Abstract. In this paper, we present an intelligent tutoring system developed to assist students in learning logic. The system helps students to learn how to construct equivalent formulas in first order logic (FOL), a basic knowledge representation language. Manipulating logic formulas is a cognitively complex and error prone task for the students to deeply understand. The system assists students to learn to manipulate and create logically equivalent formulas in a step-based process. During the process the system provides guidance and feedback of various types in an intelligent way based on user's behavior. Evaluation of the system has shown quite satisfactory results as far as its usability and learning capabilities are concerned.

Keywords: Intelligent Tutoring System, Teaching Logic, First Order Logic, Logic Equivalence

1 Introduction

The advent of the Web has changed the way that educational material and learning procedures are delivered to the students. It provides a new platform that connects students with educational resources which is growing rapidly worldwide giving new possibilities to students and tutors and offering better, cheaper and more efficient and intensive learning processes. ITSs constitute a popular type of educational systems and are becoming a fundamental mean of education delivery. Their main characteristic is that they provide instructions and feedback tailored to the learners and perform their tasks mainly based on Artificial Intelligence methods. The teacher's role is also changing and is moving from the face-to-face knowledge transmission agent to the specialist who designs the course and guides and supervises the student's learning process [10]. ITSs have been used with great success in many challenging domains to offer individualized learning to the students and have demonstrated remarkable success in helping students learn challenging content and strategies [18].

Logic is considered to be an important domain for the students to learn, but also a very hard domain to master. Many tutors acknowledge that AI and logic course contains complex topics which are difficult for the students to grasp. Knowledge Representation & Reasoning (KR&R) is a fundamental topic of Logic. A basic KR&R language is First-Order Logic (FOL), the main representative of logic-based representation languages, which is part of almost any introductory AI course and textbook. So, teaching FOL as a KR&R language is a vital aspect. Teaching and learning FOL as KR&R vehicle includes many aspects. During an AI course the student's learn to translate Natural Language (NL) text into FOL, a process also called *formalization*. A NL sentence is converted into a FOL formula, which conveys the sentence's meaning and semantics and can be used in several logic processes, such as inference and equivalency creation. Equivalency is a fundamental topic in logic. It characterizes two or more representations in a language that convey the same meaning and have the same semantics. Manipulating FOL formulas is considered to be a hard, cognitive complex and error prone process for the students to deeply understand and implement. In this paper, we present an intelligent tutoring system developed to assist students in learning logic and more specifically to help students learn how to construct equivalent formulas in FOL. The system provides interactive guidance and various types of feedback to the students.

The rest of the paper is structured as follows. Section 2 presents related work. Section 3 presents the logic equivalences in FOL. Section 4 presents the system architecture and analyzes its functionality. Section 5 presents the logic equivalent learning. More specifically describes the learning scenarios, the student's interaction and the feedback provided by the system. Section 6 presents the evaluation studies conducted and the results gathered in real classroom conditions. Finally, Section 7 concludes our work and provides directions for future work.

2 Related work

There are various systems created for teaching for helping in teaching logic [8] [19]. However, most of them deal with how to construct formal proofs, mainly using natural deduction. Logic Tutor [1] is an intelligent tutoring system (ITS) for learning formal proofs in propositional logic (PL) based on natural deduction. As an intelligent system, it adapts to the needs of the students via keeping user models. In [4], an intelligent tutoring system is developed for teaching how to construct propositional proofs and visualize student proof approaches to help teachers to identify error prone areas of the students. All the above systems, although deal with learning and/or teaching logic, they are not concerned with how to use FOL as a KR&R language.

KRRT [2] is a web–based system the main goal of which is helping students to learn FOL as a KR&R language. The student gives his/her FOL proposal sentence and the system checks its syntax and whether is the correct one. NLtoFOL [7] is a web-based system developed to assist students in learning to convert NL sentences into FOL. The student can select a NL sentence and interactively convert it in a step based approach into the corresponding FOL. In [6], we deal with teaching the FOL to CF (Clause Form) conversion, via a web-based interactive system. It provides a step-by-step guidance and help during that process. Organon [5] is a web-based tutor for basic logic courses and helps the students during practice exercises. All the above systems, although deal with learning (or teaching) logic, they do not deal with logic equivalency and how to assist students to learn how to construct logically equivalent formulas. As far as we are aware of, there is only one system that claims doing the latter. It is called IDEAS [11] and deals with rewriting formulas from propositional logic into disjunctive normal form. A student is called to transform a formula by applying one transformation rule at a time. The system provides feedback to the student. Also, the system provides a tool [12] for proving equivalences between propositional logic formulas. However, it is restricted to propositional logic and does not deal with FOL.

3 Logical Equivalences in FOL

FOL is the most widely used logic-based knowledge representation formalism. Higher order logics are difficult to handle, whereas lower order logics, such as those based on propositional calculus, are expressively poor. FOL is a KR&R language used for representing knowledge in a knowledge base, in the form of logical formulas, which can be used for automatically making inferences. Logical formulas or sentences explicitly represent properties of or relations among entities of the world of a domain. In logic, two logical formulas *p* and *q* are logically equivalent if they have the same logical content. Logical equivalence between *p* and *q* is sometimes expressed as $p \leftrightarrow q$. Logical equivalence definition in FOL is the same as in propositional logic, with the addition of rules for formulas containing quantifiers. Table 1 presents rules of logical equivalence between FOL formulas.

Table 1. Rules of logical Equivalence for FOL

Equivalence	**Name**
$p \land T \leftrightarrow p$, $p \lor F \leftrightarrow p$	Identity Laws
$p \lor T \leftrightarrow T$, $p \land F \leftrightarrow F$	Domination Laws
$p \lor p \leftrightarrow p$, $p \land p \leftrightarrow p$	Idempotent Laws
$\neg(\neg p) \leftrightarrow p$	Double Negation Law
$p \lor q \leftrightarrow q \lor p$, $p \land q \leftrightarrow q \land p$	Commutative Laws
$(p \lor q) \lor r \leftrightarrow p \lor (q \lor r)$, $(p \land q) \land r \leftrightarrow p \land (q \land r)$	Associative Laws
$(p \Rightarrow q) \leftrightarrow (\neg p \lor q)$	Implication Elimination
$\neg(p \lor q) \leftrightarrow \neg p \land \neg q$, $\neg(p \land q) \leftrightarrow \neg p \lor \neg q$	De Morgan's Laws
$\forall x\, P(x) \leftrightarrow \neg\exists x\, \neg P(x)$, $\neg\exists x\, P(x) \leftrightarrow \forall x\, \neg P(x)$	De Morgan's FOL
$p \lor (q \land r) \leftrightarrow (p \lor q) \land (p \lor r)$ $p \land (q \lor r) \leftrightarrow (p \land q) \lor (p \land r)$	Distribution Laws
$\forall x\, (P(x) \land Q(x)) \leftrightarrow \forall x\, P(x) \land \forall x Q(x)$ $\exists x\, (P(x) \lor Q(x)) \leftrightarrow \exists x\, P(x) \lor \exists x Q(x)$	Distribution Laws FOL

4 System Architecture and Function

The architecture of our system is depicted in Fig.1. It consists of five units: *Domain Model (DM), Student Model (SM), Student Interface (SI), Interface Configuration (IC)* and *Intelligent Data Unit (IDU).*

Domain Model (DM) contains knowledge related to the subject to be taught as well as the actual teaching material. It focuses on assisting students to learn how to create FOL-equivalent formulas and so syntax of FOL and equivalence rules constrains are stored in the domain model.

Student Model (SM) unit is used to record and store student related information. Also contains the system's beliefs regarding the student's knowledge of the domain and additional information about the user, such as personal information and characteristics. SM enables the system to adapt its behavior and its pedagogical decisions to the individual student who uses it [3]. Also it sketches the cognitive process that happens in the student learning sessions.

Student interface (SI) is the interactive part of the system. Through SI, a student initially subscribes to the system. During subscription, the required personal information, such as name, age, gender, year of study and email are stored. After subscription, the student can anytime access the system. SI is also responsible for configuring the interface to adapt to the needs of the specific session.

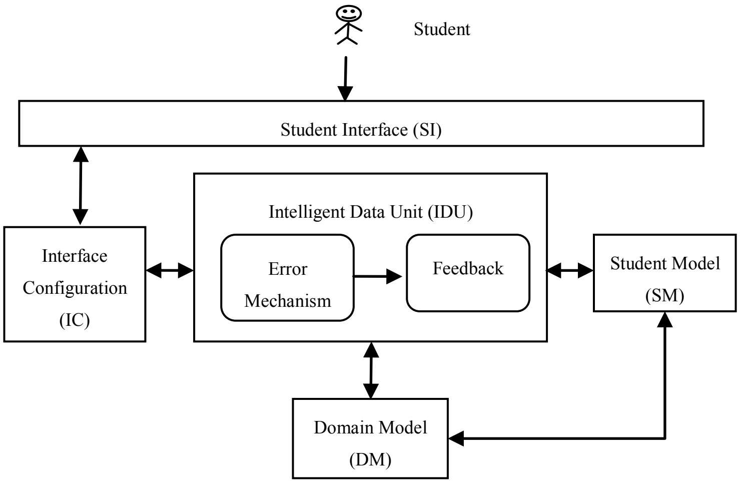


Fig.1. System architecture and its components.

Interface Configuration (IC) unit is responsible for configuring the student interface during the learning sessions, based on the guidelines given by the intelligent data unit. So, the student interface is dynamically re-configured to adapt to the needs of the specific session.

Intelligent Data Unit (IDU) interacts with IC and its main purpose is to provide guidance and feedback to the students and help during application of the logical equivalence rules. It is a rule-based system that based on the input data from user interface decides on which reconfigurations should be made to the user interface or which kind of interaction will be allowed or given to the user. It is also responsible for tracing user's mistakes and handling them in terms of appropriate feedback to the student.

IDU deals with a student's actions for each equivalence exercise as follows:

1. Let the student select an equivalence rule to apply to the FOL formula
2. Check if the selected current equivalence rule can be applied.
   - If it can, allow the student to insert his/her answer to the current rule and go to 2.
   - Otherwise, inform the student that the selected rule is not applicable, provide proper feedback and allow select a new answer.
3. Check the student's answer (formula) to the selected rule
   - If it is correct, inform his/her and go to 1.
   - Otherwise: (a) Determine the error(s) made by the student. (b) Provide feedback based on the error(s) and the corresponding equivalence rule. (c) Allow the student to give a new answer for the selected rule and go to 1.

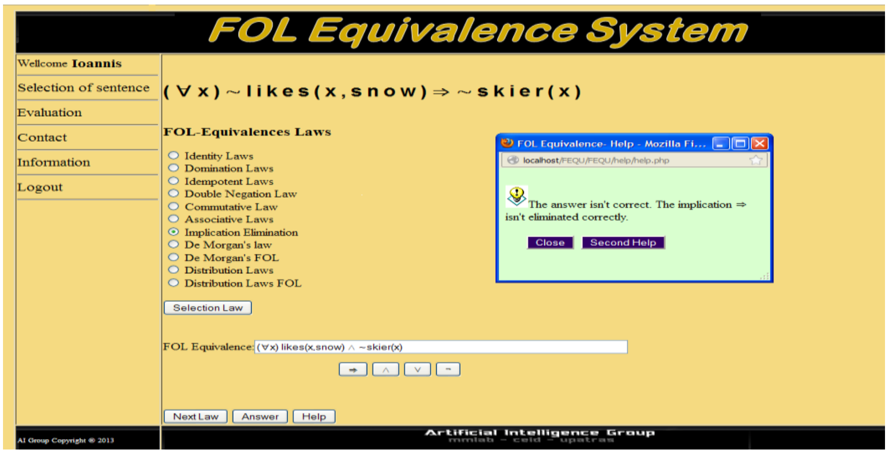
5 User interaction

The student interface of the system is dynamically reconfigured during a conversion process. After the student enters the system, he/she can select any of the existing FOL formulas/exercises and then starts its conversion into an equivalent formula. This process is made in a step-based approach where the student, at each step, has to select and implement a logic equivalence rule (see above, Table 1). At each step the student can request the system's assistance and feedback (which is based on student's actions and knowledge state). Initially, the student has to select a proper equivalence rule to implement. All the equivalence rules are presented at the working area of student's interface. The student can select a rule and apply it to the formula. If the rule cannot be applied, the system provides proper feedback messages notifying with the reason why it cannot be applied. In contrast, if the rule can be applied, a proper work area is created and the student can manipulate the formula and transform it by applying the selected rule. Then the student can submit the answer (FOL formula). After the student gives an answer, the system informs him/her whether the answer is correct or incorrect. If it is incorrect, the system performs an analysis of the student's answer to find and recognize the errors made by the student. After that the student can submit the new formula derived by the rule application.

As an example, consider the FOL formula "(∀x)~likes(x,snow) ⇒~skier(x)". Initially the student selects to apply the *implication elimination* of equivalency as illustrated in Fig. 2. The system analyses the formula and recognizes that the selected law can be applied. So, proper configurations are made on the interface and the student can insert his/her answer, which is the equivalent formula derived from the application of the rule. After the student submits his/her answer, the system analyzes it and recognizes that the implication is not removed correctly and generates the proper feedback message(s). The feedback messages are linked to the help button and the student can look at them by clicking on it.

5.1 Feedback

The behavior of the system is modeled to consist of two (feedback) loops, the inner and the outer loop respectively [16]. The main role of the inner loop is to provide feedback to the student as a reaction to his/her actions during an exercise, whereas the role of the outer loop is to select the next exercise corresponding to the student's knowledge state. The inner loop of the system is responsible for analyzing the student's answer and provides the proper feedback messages. The feedback provided, in order to enhance its effectiveness, refers to different levels of verification and elaboration. Verification concerns the confirmation whether a student's process is correct or not, while the elaboration can address the answer and related topics, discuss particular error(s) and guide the student towards the correct answer [15].



(∀x) ~likes(x,snow) ⇒ ~skier(x)

The answer isn't correct. The implication ⇒ isn't eliminated correctly.

FOL Equivalence: (∀x) likes(x,snow) ∧ ~skier(x)

Fig.2. Student Interface

The categories and the types of feedback developed are based on combinations of the classifications of feedback presented in [13] and [16]. So, the main types of feedback offered to the students by the system are the following.

- ***Minimal feedback.*** The system informs the student if the answer is correct or not.
- ***Error-specific feedback.*** When a student's answer is incorrect the system provides the proper feedback based on the errors made, indicating what makes the answer incorrect and the reason why it does it.
- ***Procedural feedback.*** The system can provide a student with hints on what has to do to correct a wrong answer and also what to do next.
- ***Bottom-out hints.*** The system can decide to give the correct answer of a step to the student. This can be done after a student's request or after constantly failure rates and circumstances.
- ***Knowledge on meta-cognition.*** The system analyzes a student's interactions and behavior and can provide meta-cognitive guiding and hints.

The system implements an incremental assistance delivery. Initially, after a student's incorrect action, starts by delivering minimal feedback, just noticing that there are errors and inconsistencies in the student's action. Error-specific feedback is offered after a student's erroneous action. Research has shown that student's motivation for understanding and learning is enhanced when errors are made [9] and the delivery of proper feedback can help the students get a deeper understanding and revise misconceptions. While a student is striving to specify the correct action, the system scales up its assistance till the delivery of the correct action/answer. Providing the correct answer in logic exercises-procedures are consider an important part of the system's assistance. Indeed, student knowledge and performance can be improved significantly after receiving knowledge of correct response feedback, indicating the correct answer [17]. The system never gives unsolicited hints to the student. If the student's answer is incorrect, the proper feedback messages are available (linked) via the help button. So, the student can get those messages on demand, by clicking on the help button. The pedagogical assumption indicates that when the student has the control of the timing of the help provided by the system, there is a greater likelihood that the help messages are received at the right time and therefore be more effective for knowledge construction [14].

6 Evaluation

We conducted an evaluation study of the system during the AI course in the fall semester of the academic year 2011-2012 at our Department. 100 undergraduate students from those enrolled in the course participated in the evaluation study. The students had already attended the lectures covering the relevant logic concepts. The methodology selected to evaluate the system is a pre-test/post-test, experimental/control group one, where the control group used a traditional teaching approach. The students were divided into two groups of 50 students each one, of balanced gender, which were named group A and group B respectively. Group A was selected to act as the experimental group and group B as the control group. Group A (experimental) did some homework through the system, whereas Group B (control) did the homework without using the system and then submit the answers to the tutor and discuss them with him.

Initially, all students took a pre-test on logical equivalence concept. The test included 15 FOL formulas-exercises and the students were asked to provide equivalent FOL formulas. After that, the students of group B were given access to the system and were asked to study for a week aiming at one 20 minutes session per day. After that intervention, the students of both groups took a final post-test including 15 FOL formulas-exercises. The two tests consisted of exercises of similar difficulty level and the score ranged from 0 to 100.

In order to analyze students' performance, an independent *t*-test was used on the pre-test. The mean and standard deviations of the pre-test were 45.18 and 14.73 for the experimental group, and 47.34 and 14.01 for the control group. As the *p*-value (Significant level) was $0.567 > 0.05$ and $t = 0.46$, it can be inferred that those two groups did not significantly differ prior to the experiment. That is, the two groups of students had statistically equivalent abilities before the experiment. In Table 2 and Table 3 the descriptive statistics and the t-test results from assessment of students' learning performance are presented. The results revealed that the mean value of the pre-test of the experimental group is higher than the mean value of the pre-test of the control group. The Levene's test confirmed the equality of variances of the control and experimental groups for pre-test ($F = 0.330$, $p = 0.567$) and post-test ($F = 3.016$, $p = 0.086$). Also the t-test result (p=0.000 < .05) shows a significant difference between the two groups. Thus, it implies that the students in the experimental group got a deeper understanding in manipulating FOL formulas and created correctly equivalent formulas for more FOL formulas exercises than the control group.

Table 2. Descriptive Statistics of Pre-test and Post-test

	Group	N	Mean	SD	SE
Pre-Test	Group A	50	45.18	14.73	2.08
	Group B	50	47.34	14.01	1.98
Post-Test	Group A	50	51.74	18.17	2.57
	Group B	50	71.56	15.43	2.18

Table 3. t-test results

Equality of variance		**F-test for variance**		**t-Test for mean**			
		F	Sig.	t	df	Sig.(2-tailed)	MD
Pre-Test	Equal	0.33	0.567	-0.751	98	0.454	-2.16
	Unequal			-0.751	97.756	0.454	-2.16
Post-Test	Equal	3.016	0.086	-5.879	98	0.000	-19.8
	Unequal			-5.879	95.49	0.000	-19.8

In the second part of the evaluation study, the students of group B, who had used the system, were asked to fill in a questionnaire. The questionnaire was made to provide both qualitative and quantitative data. It included questions for evaluating the usability of the system, asking for the students' experience and their opinions about the impact of system in learning and understanding logical equivalence. The questionnaire consisted of nine questions and the results are presented in Table 4. Questions Q1-Q6 were based on a Likert scale (1: not at all, 5: very much). Questions 7-8 were open type questions and concerned strong and weak points of the system or problems faced and also improvements that can be made to the system. Finally, question 9 was about spent time to cope with the system and had three possible answers: less than 15 min, 15-30 min and more than 30 min. Their answers show that 72% of the students needed less than fifteen minutes and only 12% of them needed more than 30 min.

Table 4. Questionnaire Results.

		ANSWERS (%)				
Q	QUESTIONS	1	2	3	4	5
1	How you rate your overall experience?	0	0	20	28	52

2	How much the system did assisted you to learn logical equivalence?	0	0	18	32	50
3	How helpful was the feedback provided?	0	4	12	36	48
4	Did you find the interface of the system helpful?	0	0	28	36	36
5	When stuck, did the system provide enough help so that you could fix the problem(s)	0	2	14	34	50
6	Do you feel more confident in dealing with logical equivalence transformations?	0	4	16	38	42

The students' answers to Q1-Q6 indicate that the majority of the them enjoyed interacting with the system and 82% of them believe that the system helped them in learning FOL equivalences. Also, 84% of them found the feedback provided by the system very useful and that assisted them in manipulating FOL formulas and creating equivalent ones.

7 Conclusions and Future Work

Logic is acknowledged by tutors to be a hard domain for students to grasp and deeply understand. It contains complex cognitive processes and students face many difficulties to understand and correctly implement them. Manipulating FOL formulas and transforming them into equivalent forms is a fundamental topic in logic, but also hard and error prone for students.

In this paper, we introduce an intelligent tutoring system developed to help students in learning how to deal with FOL equivalent formulas. It provides the student an interactive way to manipulate FOL formulas and transform them into equivalent form(s) by applying equivalence rules (or chain of rules) or proper combinations of them. The student, at each stage of the transformation, gets proper guidance and feedback by the system on his/her actions. Regarding the usefulness of the system, the reactions of the students were very encouraging. An evaluation study was conducted to test the system impact on student’s learning. The results revealed that the experimental group outperformed the control group significantly on the post-test exercises. According to the results, the students of the experimental group got a deeper understanding of the logical transformations and significantly enhanced their knowledge. Moreover, the system helped the students to improve their logic conceptual understanding and also to increase their confidence in handling equivalence.

However there are some points that the system could be improved. A direction for future research would be the development of an automatic assessment mechanism to assess the student's performance during the learning interaction with the system. This could help the system better adapt to the student.

Acknowledgements

This work was supported by the Research Committee of the University of Patras, Greece, Program “KARATHEODORIS”, project No C901.

References

1. Abraham, D., Crawford, L., Lesta, L., Merceron, A.,Yacef, K.: The Logic Tutor: A multimedia presentation. Electronic Journal of Computer-Enhanced Learning, (2001)
2. Alonso, J.A., Aranda, G.A., Martin-Matceos, F.J.: KRRT: Knowledge Representation and Reasoning Tutor. In: Moreno Diaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST LNCS, vol. 4739, pp.400–407, Springer, Heidelberg (2007)
3. Brusilovsky, P.: Student model centered architecture for intelligent learning environment. In Proc. of Fourth International Conference on User Modeling, Hyannis, MA, pp.31-36 (1994)
4. Croy, M., Barnes, T., Stamper, J.: Towards an Intelligent Tutoring System for propositional proof construction. In: Brey, P., Briggle, A., Waelbers, K. (eds.) European Computing and Philosophy Conference, pp.145–155,Amsterdam (2007)
5. Dostálová, L., Lang.J.: Organon - the web-tutor for basic logic courses Logic Journal of the IGPL 15(4), pp.305-311, (2007)
6. Grivokostopoulou, F., Perikos I., Hatzilygeroudis I.: A Web-based Interactive System for Learning FOL to CF conversion In Proc. of the IADIS International Conference e-Learning 2012, Lisbon, Portugal, pp.287-294, (2012)
7. Hatzilygeroudis, I., Perikos, I.: A web-Based Interactive System for Learning NL to FOL Conversion. New Directions in Intelligent Interactive Multimedia Systems and Services-2 Studies in Computational Intelligence, vol. 226, pp.297-307 (2009)
8. Hendriks, M., Kaliszyk, C., van Raamsdonk, F., and Wiedijk, F.: Teaching logic using a state-of-the-art proof assistant. Acta Didactica Napocensia, 3(2): 35-48, (2010)
9. Hirashima, T., Horiguchi, T., Kashihara, A., Toyoda, J.:Error-Based Simulation for Error-Visualization and Its Management. J. of Artificial Intelligence in Education, vol.9, pp.17-31 (1998).
10. Huertas, A.: Teaching and Learning Logic in a Virtual Learning Environment. Logic Journal of the IGP 15(4), pp.321–331 (2007)
11. Lodder, J., Passier, H., Stuurman, S.:Using IDEAS in teaching logic, lessons learned. Computer Science and Software Engineering, In Proc. of International Conference Computer Science and Software Engineering, vol. 5, pp.553–556 (2008).
12. Lodder, J., Heeren, B.: A teaching tool for proving equivalences between logical formulae. In Third International Congress on Tools for Teaching Logic, pp.66-80, (2011).
13. Narciss, S.: Feedback strategies for interactive learning tasks. In J. M. Spector, M. D. Merrill, J. J. G. Van Merrienboer, M. P. Driscoll (Eds.), Handbook of research on educational communications and technology 3rd ed., pp.125-143,(2008)
14. Renkl, A., Atkinson, R. K., Maier, U. H., Staley, R.:From example study to problem solving: Smooth transitions help learning. J. of Experimental Education,vol.70, pp.293–315, (2002).
15. Shute, V.J. :Focus on formative feedback, Review of Educational Research, Vol. 78, No. 1, pp.153–189 (2008).
16. VanLehn, K.: The behavior of tutoring systems. International Journal of Artificial Intelligence in Education, 16,pp.227-265
17. Wang, S-L., Wu, P-Y.: The role of feedback and self-efficacy on Web-based learning: the social cognitive perspective, Computers & Education vol.51 pp.1589-1598(2008).
18. Woolf, B. P. :Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning. Burlington MA: Morgan Kaufman Publishers (2009).
19. Sieg, W., Scheines, R.: Computer Environments for Proof Construction. Interactive Learning Environments 4(2), pp. 159–169 (1994).

Informing the Design of a Game-Based Learning Environment for Computer Science: A Pilot Study on Engagement and Collaborative Dialogue

Fernando J. Rodríguez, Natalie D. Kerby, Kristy Elizabeth Boyer

Department of Computer Science, North Carolina State University, Raleigh, NC 27695
{fjrodri3,ndkerby,keboyer}@ncsu.edu

Abstract. Game-based learning environments hold great promise for supporting computer science learning. The ENGAGE project is building a game-based learning environment for middle school computational thinking and computer science principles, situated within mathematics and science curricula. This paper reports on a pilot study of the ENGAGE curriculum and gameplay elements, in which pairs of middle school students collaborated to solve game-based computer science problems. Their collaborative behaviors and dialogue were recorded with video cameras. The analysis reported here focuses on nonverbal indicators of disengagement during the collaborative problem solving, and explores the dialogue moves used by a more engaged learner to repair a partner's disengagement. Finally, we discuss the implications of these findings for designing a game-based learning environment that supports collaboration for computer science.

Keywords: Engagement, Collaboration, Dialogue, Game-Based Learning.

1 Introduction

Supporting engagement within computer science (CS) education is a central challenge for designers of CS learning environments. More broadly, engagement is a subject of increasing attention within the AI in Education community. A growing body of empirical findings has revealed the importance of supporting learner engagement. Particular forms of disengagement have been associated with decreased learning, both overall and with respect to local learning outcomes within spoken dialogue tutoring systems [1, 2]. Targeted interventions can positively impact engagement; for example, metacognitive support may influence students to spend more time on subsequent problems, and integrating student performance measures into a tutoring system allows them to reflect on their overall performance [3]. A promising approach to support engagement involves adding game elements to intelligent tutors or other learning environments [4, 5] or creating game-based learning environments with engaging narratives [6]; both approaches have been shown to increase student performance and enjoyment in general. However, even with these effective systems, some disengaged behaviors are negatively associated with learning, and the relationships between engagement and learning are not fully understood.

Collaboration is another promising approach for supporting engagement and can be combined with game-based learning environments [7]. Results have demonstrated the importance of well-timed help for collaborators [8] and the promise of pedagogical agents that support self-explanation [9]. This study considers collaboration in the problem-solving domain of computer science, where a combination of hints and collaboration support may be particularly helpful [10]. However, many questions remain regarding the best sources and types of engagement support in this context.

Game-based learning environments for teaching computer science have started to become popular in recent years. The CodeSpells game [11] aims to teach middle school students how to program in the Java programming language. The ENGAGE project aims to develop, implement, and evaluate a narrative-centered, game-based learning environment that will be deployed in middle school for teaching computer science principles. The game is designed to be played collaboratively by pairs of students. Presently, the project is in its design and implementation phase, conducting iterative refinement and piloting of curriculum and gameplay elements. During this process, we aim to extract valuable lessons about how middle school students collaborate to solve computer science problems and how this collaboration can be supported within an intelligent game-based learning environment.

This paper reports on a pilot study of the ENGAGE curriculum and simulated gameplay elements. In this study, pairs of students collaborated and their collaborative behaviors and dialogue were recorded with video cameras. Nonverbal indicators of disengagement were annotated manually across the videos. We report an analysis of these disengagement behaviors by students' collaborative role, and explore the dialogue moves used by a more engaged learner to repair a partner's disengagement.

2 ENGAGE Game Based Learning Environment

The main goals of the ENGAGE project, which is currently in its design and implementation phase, are to create a highly engaging educational tool for teaching computer science to middle school students, contribute to research on the effectiveness of game-based learning, and investigate its potential to broaden participation of underrepresented groups in computer science. During the first year of the project, the first draft of the curriculum to be used within the environment was developed. The curriculum is based on the CS Principles course under development by the College Board [12] with the goal of shifting focus from a specific programming language (Java, in the case of the existing AP Computer Science course) to the broader picture of computer science concepts. The CS Principles curriculum emphasizes seven *big ideas*:

Table 1. CS Principles focused evidence statement examples

CS Principles Number	Evidence Statement
6b	Explanation of how number bases, including binary and decimal, are used for reasoning about digital data
13a	Explanation of how computer programs are used to process information to gain insight and knowledge
18b	Explanation of how an algorithm is represented in natural language, pseudo-code, or a visual or textual programming language
24a	Use of an iterative process to develop a correct program
30c	Explanation of how cryptography is essential to many models of cybersecurity

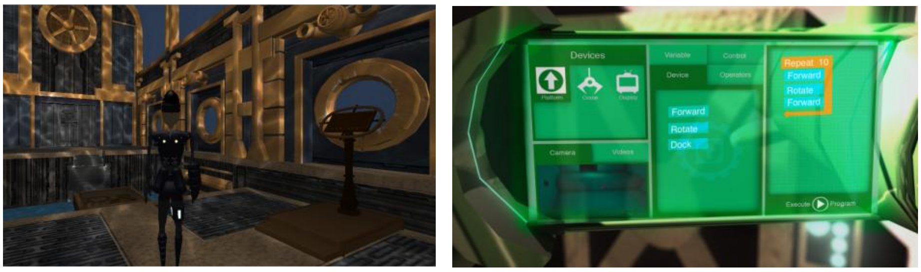
1. Computer science is a creative process
2. Abstractions can reduce unimportant details and focus on relevant ones
3. Big data can be analyzed using various techniques in order to create a new understanding or refine existing knowledge
4. Algorithms are a sequence of steps used to solve a problem and can be applied to structurally similar problems
5. These algorithms can be automated using a programming language
6. The Internet has revolutionized communication and collaboration
7. Computer science has an impact on the entire world

Through an iterative process, we selected a subset of the CS Principles curriculum by analyzing the evidence statements [13] for suitability within a game-based learning environment and for appropriateness for the middle school audience. Additional validation of this curriculum will be undertaken by middle school teachers during an upcoming summer institute and through pilot testing. An example of learning objectives to be implemented as game-based learning activities is shown in Table 1.

The setting of the game is an underwater research facility that has been taken over by a rogue scientist. Students take on the role of a computer scientist sent to investigate the situation, reconnect the station's network, and thwart the villain's plot by solving various computer science puzzles in the form of programming tasks. There are two main gameplay mechanics: players can move around in the 3D environment in a similar manner to many 3D platforming games (Fig. 1a), and different devices within the environment can be programmed using a visual programming interface. Players can drag "blocks" that represent programming functions and stack them together to create a program (Fig. 1b).[1] By programming these devices in certain ways, players can manipulate the environment and solve each in-game area's puzzle and move on to the next task. The game sections are divided into four main levels:

[1] This drag-and-drop programming language with blocks is closely modeled after and inspired by Scratch [21], but for compatibility reasons, a customized programming environment is being created for the ENGAGE game.

- *Tutorial*: Students are introduced to the game environment and shown how to use the controls for both gameplay mechanics. They are also given an overview of basic programming concepts (sequences of statements, loops, and conditionals), as well as the concept of broadcasting (sending signals from one device to another).
- *Digital World*: The puzzles in this level involve binary numbers, and students must convert these binary numbers into an understandable form (decimal, text, color image, etc.) in order to solve them and progress. The conceptual objective of this level is that computers communicate in binary and that the meaning behind a binary sequence depends on its interpretation.
- *Cybersecurity*: Before the students can reconnect the station's network, they must establish proper cybersecurity measures so that their communications are not compromised by the villain. In this level, students learn about cryptography and various encryption techniques in order to ensure a safe network connection.
- *Big Data*: The research station that students must restore had been studying different aspects of the undersea environment, including the pollution of the water and how it has affected the life forms that inhabit the area. Students must try to reason about this data by performing basic analyses and creating visual models that are embodied in the 3D environment, which will enable students to progress to the final level.

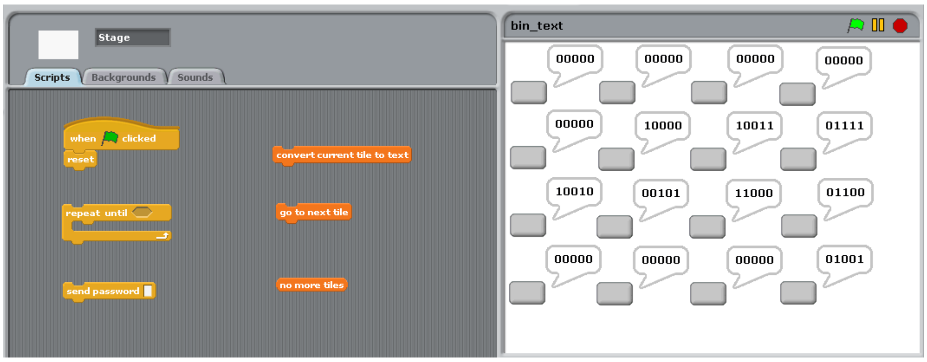


a) Game environment b) Programming interface

Fig. 1. Engage screenshots

3 Pilot Study

The pilot study was conducted within a computer science elective course for middle school students (ages 11 to 14) at a charter middle school. Students attended 4 sessions lasting between 90 and 120 minutes long, facilitated by members of the ENGAGE project team. Each of the sessions resulted in a corpus of data, though only one of these, which involves solving a binary puzzle within a visual programming environment, is analyzed within this paper. This paper focuses on the final session of the course, in which students worked in pairs to solve three game-based tasks using Build Your Own Blocks, a drag-and-drop visual programming language [14]. This activity simulates programming within ENGAGE, whose programming environment is still under development. Student participants included 18 males and 2 females, though the female pair was absent on the day the present corpus was collected. This gender disparity is an intrinsic problem in many technology electives and is an important consideration of the ENGAGE project, which will examine differential outcomes for students from underrepresented groups using the game-based learning environment.



00000	00000	00000	00000
00000	10000	10011	01111
10010	00101	11000	01100
00000	00000	00000	01001

Fig. 2. Visual programming interface for final problem in pilot study

The exercises the students solved involved converting binary numbers into decimal numbers and textual characters. The first problem asked the students to write a program that would convert the given binary numbers into decimal numbers and highlight the cells that contained even numbers. The next problem was identical to the first except that the students had to highlight the prime numbers instead. The final and most challenging problem asked students to convert the binary numbers into textual characters, manually decipher a password, and input this password into their program (Fig. 2). All subroutines corresponding to binary conversions, highlighting blocks, number comparisons, and password entry were provided as blocks that students could use within their own programs. This design choice was made to abstract some complex implementation from students so they could focus on planning and implementing the steps to solve their problem.

For the duration of the exercise, students collaborated in pairs and took turns controlling the keyboard and mouse on a single computer [7]. In computer science education, this paradigm is referred to as pair programming: the *driver* actively creates the solution, while the *navigator* provides feedback [15] (Fig. 3). Research has shown that pair programming can provide many benefits to college-level students taking introductory programming courses, especially those with little to no prior programming experience. A review by Preston [16] highlights some benefits: students create higher-quality programs as a result of the communication of ideas between partners; they can achieve a better understanding of programming by supporting each other through the exercise; and although the activity is collaborative, individual test scores and course performance are also improved.

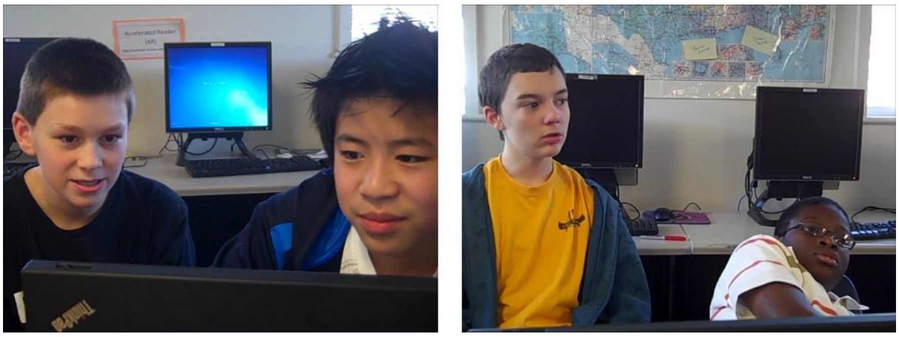
Students were asked by a researcher to switch roles every six minutes. When a pair would finish a task, they were asked to raise their hands and wait for a researcher to verify their program solution. If it was correct, the researcher would verbally describe the next exercise and set up the programming interface; if not, the researcher would provide general feedback on the proposed solution. Students were also allowed to raise their hands if they had any questions for the researcher regarding the programming task. The allotted time to complete all three programming tasks was 40 minutes, with two pairs completing them sooner.

4 Video Corpus and Disengagement Annotation

During this pilot study, video was recorded for all nine pairs of students using a tripod-mounted digital camera recording at 640x480 resolution and 30 frames per second. The nine videos were divided into 5-minute segments to facilitate annotation and analysis. Of the total 65 segments, 25 were randomly selected for annotation and serve as the basis for the results presented here (a subset was necessary due to the time requirement of manual annotation, in this case approximately 8 minutes per minute of video). Each segment was manually annotated by a judge for student disengagement by observing for one of three signs of disengagement. First, *posture* was considered to indicate disengagement when gross postural shifts clearly suggested that the student was attending to something other than the programming task; this exaggerated disengaged posture was often accompanied by other indications of disengagement such as off-task speech (Fig. 3b). Another indicator of disengagement was averted *gaze*, which commonly accompanied the other two signs but could occur independently. Finally, students would sometimes engage in off-task *dialogue* with their partners, or even with other students in the classroom. It is important to note that we do not equate off-task behavior with disengagement; there were instances in which students continued to work on the learning task while holding off-task conversations with their partners. Sabourin et al. [17] show that off-task behavior can be a way for students to cope with negative emotions, such as confusion or frustration. Likewise, student disengagement does not necessarily imply off-task behavior. Disengagement in this context is defined primarily as focusing attention on something other than the learning task; identifying cognitive and affective states underlying the disengagement is left to future analyses.

To annotate the videos, each human judge would watch until disengagement was observed by either of the two collaborators, paused the video, annotated the start time of the disengagement event, then continued and annotated the end time, returning to previous points of the video as needed. Judges thus marked episodes of disengagement, as well as who appeared to facilitate re-engagement: did the student shift her attention back to the programming task by herself?; did the student's partner ask for her assistance?; or did an instructor need to arrive to provide feedback or clarify any questions? In order to establish reliability of this annotation scheme, 12 of the 65 video segments were randomly selected and assigned to two judges, and the tagged segments were discretized into one-second intervals in which each student was classified as either engaged or disengaged by each judge. The Kappa for disengagement was 0.59 (87.25% agreement). In other words, approximately 87.25% of the time, both judges applied the same engagement tag. Adjusting for chance (that is, students were more likely to be engaged than disengaged at a given point) the

Kappa agreement statistic was 0.59, indicating fair agreement [18]. For the events on which both judges agreed that disengagement had occurred, the tag for who facilitated re-engagement resulted in 78.57% agreement, or a Kappa of 0.60, indicating moderate agreement.



a) Engaged pair

b) Disengaged pair

Fig. 3. Collaborative setup

5 Results

Overall, student drivers spent an average of 16.4% of their time disengaged (st. dev.=16.6%), compared to a much higher 42.6% for navigators (st. dev.=24.1%). This is not surprising, since drivers are more actively engaged in the programming activity. Across both roles, out of the student re-engagement events, 76.8% were annotated as self re-engagements, with the remaining events corresponding to an external source of re-engagement (either partner or instructor). However, the collaborative role plays an important part in self re-engagement: drivers had an 87.7% probability of self re-engaging, while navigators had a lower 68.7% probability of self re-engaging. These findings may indicate that repairing one's own disengaged state is more challenging for the collaborative partner who is not actively at the controls.

We examine instances in which annotators marked that the driver re-engaged a disengaged navigator through dialogue. There are 22 such instances. Four are questions addressed to the collaborative partner, such as, "OK, now where?" and "Do we delete this?" These questions re-engaged the navigator in part because attending to the speaker's questions is a social dialogue norm. Two utterances served as exclamations, e.g., "What the heck?" In these cases, the driver was expressing surprise with an event in the learning environment, which drew the disengaged student's attention back to the task. The remaining utterances were fragments, such as, "Pick up current tile…," though one utterance explicitly reminded the disengaged student about short time remaining, "So we only have a couple of minutes."

To examine these re-engagement events in context, two excerpts are considered (Table 2). In Excerpt A, the navigator gets stuck and raises his hand to ask for help when he notices the instructor is nearby, briefly becoming disengaged while his partner continues to work on the exercise. The driver then turns to his partner and asks for feedback, causing the navigator to re-engage into the learning activity. In Excerpt B, the students had just received feedback on from the instructor when the navigator engages in off-topic dialogue with another team. Meanwhile, the driver makes a plan and then calls for the navigator's attention, re-engaging him.

Table 2. Dialogue excerpts featuring navigator disengagement

Timestamp	Role	Dialogue Excerpt A
19:25	**Navigator:**	OK, if prime, number is prime. Dang!
		[*Navigator notices instructor nearby, raises hand*]
19:34	**Navigator:**	Uh...
Disengaged		[*Navigator looks away from screen, leans back on seat*]
19:38	**Driver:**	OK, now where?
Re-engaged		[*Navigator points at program block*]
19:40	**Navigator:**	Put it there.
		Dialogue Excerpt B
		[Note: students are discussing '@' symbols]
26:01	**Navigator:**	OK, @'s. Do you want more @'s... (inaudible)
26:08	**Driver:**	One two three four five
Disengaged		[*Navigator looks away to talk to another student*]
26:14	**Driver:**	I have an idea. You (taps navigator's shoulder)
Re-engaged		
26:16	**Navigator:**	Me?

6 Discussion

These excerpts suggest that within a collaborative game-based learning environment for computer science, providing both students with a sense of control may be particularly important. Because it may be more difficult to stay engaged on a task if one is not actively participating in it, particularly for younger audiences, the issue of mutual participation is paramount within the learning environment. The narrative game-based learning framework may prove particularly suitable for addressing this challenge: drivers and navigators can be provided with separate responsibilities and even with complementary information so that the participation of both students is required to complete the game-based tasks. Examples include designing the algorithmic solution to the problem or performing some calculations relevant to the main task; Williams and Kessler [19] state that 90% of students surveyed about pair programming listed these as the tasks that the navigator typically assists with. These may, in turn, help the navigator experience a heightened sense of control, and thereby, engagement.

This pilot study demonstrated that because of strong social norms associated with human dialogue, strategic moves by a partner can serve to re-engage a student. Typically, drivers will ask their partners for feedback if they are unsure of their solution or if they are inexperienced programmers. In these cases, an active conversation between both students occurs, and both students are engaged. An intelligent game-based learning environment that senses disengagement may be able to scaffold this type of dialogue in order to mitigate disengagement on the part of either student.

7 Conclusion and Future Work

Game-based learning environments hold great promise for supporting computer science education. The ENGAGE project is developing a game-based learning environment for middle school computer science, and we have presented results from an early pilot study for the curriculum and some simulated elements of gameplay in which students worked collaboratively in pairs to solve computer science problems. The results suggest that supporting engagement may be particularly important within a collaborative situation for the student who is not at the controls. Providing both students with an active role during gameplay, and scaffolding dialogue to re-engage a student who has become disengaged, are highly promising directions for intelligent game-based learning environments. Both of these interventions would be well supported within a narrative-centered, game-based learning environment framework.

There are several important directions for future work regarding engagement within game-based learning environments for computer science. First, the current study was very limited in sample size and diversity of participants, so expanding the scope of students considered is a key consideration. It is also important to examine the duration of engagement once re-engagement has occurred and the effectiveness of interventions with respect to longer-term engagement. Additionally, in contrast to the fully manual video annotation presented here, it would be beneficial to integrate automated methods of measuring disengagement, such as the ones presented by Arroyo and colleagues [20]. Finally, addressing issues of diversity and groupwise differences of re-engagement strategies is an essential direction in order to develop game-based learning environments that support engagement and effective learning for all students.

Acknowledgements. This paper relies on the contributions of the entire ENGAGE research team, including James Lester, Bradford Mott, Eric Wiebe, Rebecca Hardy, Wookhee Min, Kirby Culbertson, and Marc Russo. The authors wish to thank Joseph Grafsgaard and Alexandria Vail for their helpful contributions. This work is supported in part by NSF through grants CNS-1138497 and CNS-1042468. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the participants and do not necessarily represent the official views, opinions, or policy of the National Science Foundation.

References

1. Forbes-Riley, K., Litman, D.: When Does Disengagement Correlate with Learning in Spoken Dialog Computer Tutoring? Proceedings of AIED. pp. 81–89 (2011).
2. Cocea, M., Hershkovitz, A., Baker, R.S.J.: The Impact of Off-Task and Gaming Behaviors on Learning: Immediate or Aggregate? Proceedings of AIED. pp. 507–514 (2009).

3. Arroyo, I., Ferguson, K., Johns, J., Dragon, T., Meheranian, H., Fisher, D., Barto, A., Mahadevan, S., Woolf, B.P.: Repairing Disengagement With Non-Invasive Interventions. Proceedings of AIED. pp. 195–202 (2007).
4. Jackson, G.T., Dempsey, K.B., McNamara, D.S.: Short and Long Term Benefits of Enjoyment and Learning within a Serious Game. Proceedings of AIED. pp. 139–146 (2011).
5. Rai, D., Beck, J.E.: Math Learning Environment with Game-Like Elements: An Incremental Approach for Enhancing Student Engagement and Learning Effectiveness. Proceedings of ITS. pp. 90–100 (2012).
6. Rowe, J., Shores, L., Mott, B., Lester, J.C.: Integrating Learning, Problem Solving, and Engagement in Narrative-Centered Learning Environments. *IJAIED*. 21, 115–133 (2011).
7. Meluso, A., Zheng, M., Spires, H.A., Lester, J.C.: Enhancing 5th Graders' Science Content Knowledge and Self-Efficacy through Game-Based Learning. Computers & Education Journal. 59, 497–504 (2012).
8. Chaudhuri, S., Kumar, R., Howley, I., Rosé, C.P.: Engaging Collaborative Learners with Helping Agents. Proceedings of AIED. pp. 365–272 (2009).
9. Hayashi, Y.: On Pedagogical Effects of Learner-Support Agents. Proceedings of ITS. pp. 22–32 (2012).
10. Holland, J., Baghaei, N., Mathews, M., Mitrovic, A.: The Effects of Domain and Collaboration Feedback on Learning in a Collaborative Intelligent Tutoring System. Proceedings of AIED. pp. 469–471 (2011).
11. Esper, S., Foster, S.R., Griswold, W.G.: On the Nature of Fires and How to Spark Them When You're Not There. Proceedings of the SIGCSE Conference. pp. 305–310. ACM Press, New York, New York, USA (2013).
12. Stephenson, C., Wilson, C.: Reforming K-12 Computer Science Education… What Will Your Story Be? ACM Inroads. 3, 43–46 (2012).
13. The College Board: AP CS Principles: Learning Objectives and Evidence Statements, (2010).
14. Harvey, B., Mönig, J.: Bringing "No Ceiling" to Scratch: Can One Language Serve Kids and Computer Scientists? Constructionism. pp. 1–10 (2010).
15. Nagappan, N., Williams, L., Ferzli, M., Wiebe, E., Miller, C., Balik, S., Yang, K.: Improving the CS1 Experience with Pair Programming. Proceedings of the SIGCSE Conference. pp. 359–362 (2003).
16. Preston, D.: Pair Programming as a Model of Collaborative Learning: A Review of the Research. Journal of Computing Sciences in Colleges. 20, 39–45 (2005).
17. Sabourin, J., Rowe, J.P., Mott, B.W., Lester, J.C.: When Off-Task is On-Task: The Affective Role of Off-Task Behavior in Narrative-Centered Learning Environments. Proceedings of AIED. pp. 523–536 (2011).
18. Landis, J.R., Koch, G.G.: The Measurement of Observer Agreement for Categorical Data. Biometrics. 33, 159–174 (1977).
19. Williams, L.A., Kessler, R.R.: All I Really Need to Know about Pair Programming I Learned in Kindergarten. Communications of the ACM. 5, 108–114 (1999).
20. Arroyo, I., Cooper, D.G., Burleson, W., Woolf, B.P., Muldner, K., Christopherson, R.: Emotion Sensors Go to School. Proceedings of AIED. pp. 17–24 (2009).
21. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., Kafai, Y.: Scratch: Programming for All. Communications of the ACM. 52, 60–67 (2009).

When to Intervene: Toward a Markov Decision Process Dialogue Policy for Computer Science Tutoring

Christopher M. Mitchell, Kristy Elizabeth Boyer, and James C. Lester

Department of Computer Science, North Carolina State University,
Raleigh, North Carolina, USA
{cmmitch2, keboyer, lester}@ncsu.edu

Abstract. Designing dialogue systems that engage in rich tutorial dialogue has long been a goal of the intelligent tutoring systems community. A key challenge for these systems is determining when to intervene during student problem solving. Although intervention strategies have historically been hand-authored, utilizing machine learning to automatically acquire corpus-based intervention policies that maximize student learning holds great promise. To this end, this paper presents a Markov Decision Process (MDP) framework to learn when to intervene, capturing the most effective tutor turn-taking behaviors in a task-oriented learning environment with textual dialogue. This framework is developed as a part of the JavaTutor tutorial dialogue project and will contribute to data-driven development of a tutorial dialogue system for introductory computer science education.

Keywords: Tutorial Dialogue, Markov Decision Processes, Reinforcement Learning

1 Introduction

The effectiveness of tutorial dialogue has been widely established [1, 2]. Today's tutorial dialogue systems have been successful in producing learning gains as they support problem solving [3–5], encourage collaboration [6, 7], and adapt to student responses [8]. These systems have also been shown to be successful in implementing some affective adaptations of human tutors [5, 9]. Recent research into tutorial dialogue systems with unrestricted turn-taking has shown promise for simulating the natural tutorial dialogue interactions of a human tutor [7]. Recognizing and simulating the natural conversational turn-taking behavior of humans continues to be an area of active research [10–12], and there has recently been renewed interest in developing dialogue systems that harness unrestricted turn-taking paradigms [7, 13, 14].

The JavaTutor tutorial dialogue project aims to build a tutorial dialogue system with unrestricted turn-taking and rich natural language to support introductory computer science students. The overarching paradigm of this project is to automatically derive tutoring strategies using machine learning techniques applied to a corpus collected from an observational study of human-human tutoring. In particular, the project focuses on how to devise tutorial strategies that deliver both cognitive and affective scaffolding in the most effective way. The project to date has seen the collection of a large corpus of tutorial dialogue featuring six repeated interactions with tutor-student pairs, accompanied by data on learning and attitude for each session as well as across the study [15–17]. This paper describes an important first step toward deriving tutorial dialogue policies automatically from the collected corpus in a way that does not simply mimic the behavior of human tutors, but seeks to identify the most effective tutorial strategies and implement those within the system's dialogue policy.

In recent years, reinforcement learning (RL) has proven useful for creating tutorial dialogue system policies in structured problem-solving interactions, such as what type of question to ask a student [18] and whether to elicit or tell the next step in the solution [19]. In order to harness the power of RL-based approaches within a tutorial dialogue system for computer science education, two important research problems must be addressed. First, a representation must be formulated in which student computer programming actions, which can occur continuously or in small bursts, can be segmented at an appropriate granularity and provided to the model. Second, because student dialogue moves, tutor dialogue moves, and student programming actions can occur in an interleaved manner with some overlapping each other, features to define the Markov Decision Process state space must be identified that preserve the rich, unrestricted turn-taking and mixed-initiative interaction to the greatest extent possible. In a first effort to address these challenges, this paper presents a novel application of RL-based approaches to the JavaTutor corpus of textual tutorial dialogue. In particular, the focus here is automatically learning when to intervene from this fixed corpus of human-human task-oriented tutorial dialogue with unrestricted turn-taking. The presented approach and policy results can inform data-driven development of tutorial systems for computer science education.

2 Human-Human Tutorial Dialogue Corpus

To date, the JavaTutor project has seen the collection of an extensive corpus of human-human tutoring. Between August 2011 and March 2012, 67 students interacted with experienced tutors through the Java Online Tutoring Environment (Figure 1). Students were drawn from a first-year engineering course on a voluntary basis. They earned partial course credit for their participation. Students who reported substantial programming experience in a pre-survey were excluded from the experienced-tutoring condition (and were instead placed in a peer-tutoring collaborative condition that is beyond the scope of this paper), since the target population of the JavaTutor tutorial dialogue system is students with no programming experience. Each student completed six tutoring sessions over a period of four weeks, and worked with the same tutor for all interactions. Each tutoring session was limited to forty minutes.

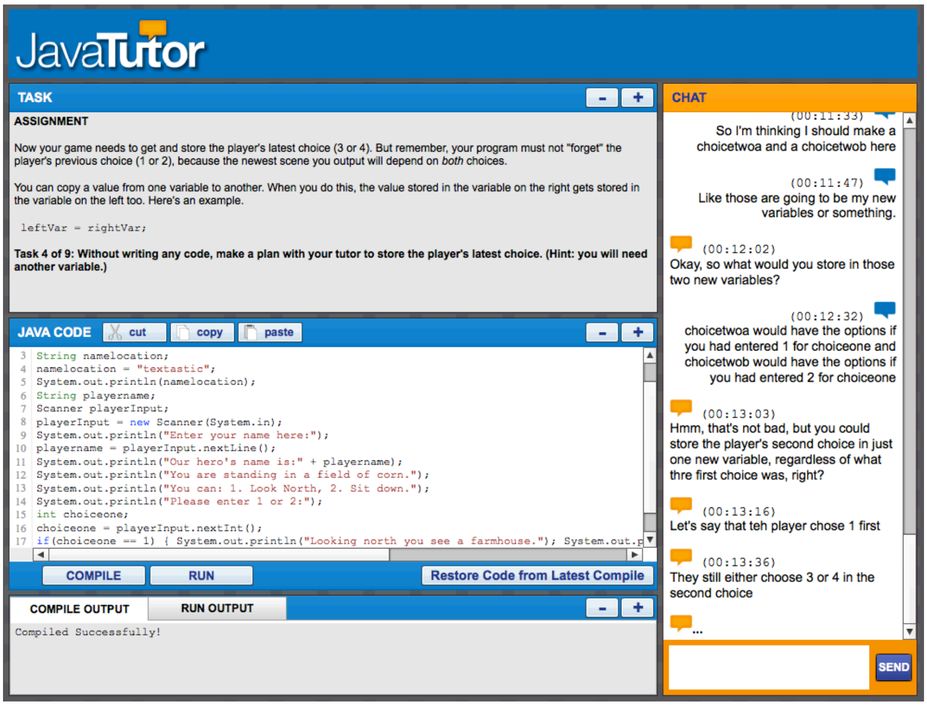
Seven tutors participated in the study. Their experience level ranged from multiple years' experience in one-on-one tutoring to one semester's experience as a teaching assistant or small group tutor. Gender distribution of the tutors was three female and four male. Tutors were provided with printed learning objectives for each session and were reminded that they should seek to support the students' learning as well as motivational and emotional state. Also, because each subsequent tutoring session built on the completed computer program from the preceding session, tutors were encouraged to ensure that students completed the required components of the programming task within the allotted forty-minute time frame.

The overarching computer science problem-solving task was for students to create a text-based adventure game in which a player can explore scenes based on menu choices. In order to implement the adventure game, students learned a variety of programming concepts and constructs. This paper focuses on the first of the six tutoring sessions. The learning objectives covered in this first session included compiling and running code, writing comments, variable declaration, and system I/O. For each learning objective, there was a conceptual component and an applied component. For example, for the learning objective related to compiling code, the conceptual learning objective was for students to explain that compilation translates human-readable Java programs into machine-readable forms. The applied learning objective was for students to demonstrate that they can compile a program by pressing the "compile" button within the interface.

The Java Online Learning Environment, shown in Figure 1, supports textual dialogue between the human tutor and student. It also provides tutors with a real-time synchronized view of the student's workspace. The interface allows for logging events to a database with millisecond precision, making it straightforward to reconstruct the events of a session from these logs. There are two information channels between a tutor and a student. The first of these, the messaging pane, supports unrestricted textual dialogue between a tutor and a student, similar to common instant messaging applications. There are no restrictions placed on turn-taking, allowing either person to compose a message at any time. In addition, both students and tutors are notified when their partner is composing a message. The second information channel is the student's workspace. A tutor can see progress on the Java program written by the student in real-time, but the tutor is not able to edit the program directly. The Java programming environment is scaffolded for novices: it hides class declarations, method declarations, and import statements from the student, lowering the amount of complex syntax visible. Students effectively compose their programs within a main method "sandbox".

In order to measure the effectiveness of each session, students completed a pre-test at the beginning of each session and a post-test at the end of each session evaluating their knowledge of the material to be taught in that lesson. From these, we computed normalized learning gain using the following equation:

$$normalizedLearningGain = \begin{cases} \dfrac{posttest - pretest}{1 - pretest}, & posttest > pretest \\ \dfrac{posttest - pretest}{pretest}, & posttest \leq pretest \end{cases} \tag{1}$$



ASSIGNMENT

Now your game needs to get and store the player's latest choice (3 or 4). But remember, your program must not "forget" the player's previous choice (1 or 2), because the newest scene you output will depend on *both* choices.

You can copy a value from one variable to another. When you do this, the value stored in the variable on the right gets stored in the variable on the left too. Here's an example.

```
leftVar = rightVar;
```

Task 4 of 9: Without writing any code, make a plan with your tutor to store the player's latest choice. (Hint: you will need another variable.)

```java
String namelocation;
namelocation = "textastic";
System.out.println(namelocation);
String playername;
Scanner playerInput;
playerInput = new Scanner(System.in);
System.out.println("Enter your name here:");
playername = playerInput.nextLine();
System.out.println("Our hero's name is:" + playername);
System.out.println("You are standing in a field of corn.");
System.out.println("You can: 1. Look North, 2. Sit down.");
System.out.println("Please enter 1 or 2:");
int choiceone;
choiceone = playerInput.nextInt();
if(choiceone == 1) { System.out.println("Looking north you see a farmhouse."); System.out.p
```

```
Compiled Successfully!
```

(00:11:33)
So I'm thinking I should make a choicetwoa and a choicetwob here

(00:11:47)
Like those are going to be my new variables or something.

(00:12:02)
Okay, so what would you store in those two new variables?

(00:12:32)
choicetwoa would have the options if you had entered 1 for choiceone and choicetwob would have the options if you had entered 2 for choiceone

(00:13:03)
Hmm, that's not bad, but you could store the player's second choice in just one new variable, regardless of what thre first choice was, right?

(00:13:16)
Let's say that teh player chose 1 first

(00:13:36)
They still either choose 3 or 4 in the second choice

...

Fig. 1. A student's view of the JavaTutor human tutoring interface

This equation, adapted from Marx and Cummings [20] allows for the possibility of negative learning gain during a session, a phenomenon that occurred three times in the corpus. These normalized learning gain values can range from -1 to 1. In the present study normalized learning gains ranged from -0.29 to 1 (mean = 0.42; median = 0.45; st. dev. = 0.32). Students scored significantly higher on the post-test than the pre-test ($p < .001$).

3 Building the Markov Decision Process

The goal of the analysis presented here is to derive an effective tutorial intervention policy—*when* to intervene—from a fixed corpus of student-tutor interactions. From the tutors' perspective, the decision to intervene was made based on the state of the interaction as observed through the two information channels in the interface: the textual dialogue pane and the synchronized view of the student's workspace. In order to use a MDP framework to derive an effective intervention policy, we describe a representation of the interaction state as a collection of features from these information channels.

A Markov Decision Process is a model of a system in which a policy can be learned to maximize reward [21]. It consists of a set of states *S*, a set of actions *A* representing possible actions by an agent, a set of transition probabilities indicating how likely it is for the model to transition to each state $s' \in S$ from each state $s \in S$ when the agent performs each action $a \in A$ in state s, and a reward function R that maps real values onto transitions and/or states, thus signifying their utility.

The goal of this analysis is to model tutor interventions during the task-completion process, so the possible actions for a tutor were to intervene (by composing and sending a message) or not to intervene. Hence, the set of actions is defined as $A = \{TutorMove, NoMove\}$. We chose three features to represent the state of the dialogue, with each feature taking on one of three possible values. These features, described in Table 1, combine as a triple to form the states of the MDP as (Current Student Action, Task Trajectory, Last Action). These three features were chosen because they succinctly represent the current state of the dialogue in terms of turn-taking information in the *Current Action* and *Last Action* features, while the recent behavior of the student is captured in the *Task Trajectory* and *Current Action* features. Thus, these features supply an agent with sufficient information to learn a basic intervention policy while relying only on automatically annotated features. By selecting a small state space and action space, we avoid data sparsity issues [22], thereby decreasing the likelihood of states being insufficiently explored in our corpus, and increasing the likelihood of producing a meaningful intervention policy.

Table 1. The features that define the states of the Markov Decision Process

Current Student Action	Task Trajectory	Last Action
• ***Task***: Working on the task • ***StudentDial**:* Writing a message to the tutor • ***NoAction***: No current student action	• ***Closer***: Moving closer to the final correct solution • ***Farther***: Moving away from correct solution • ***NoChange***: Same distance from correct solution	• ***TutorDial***: Tutor message • ***StudentDial***: Student message • ***Task*:** Student worked on the task

In addition, the model includes 3 more states: an *Initial* state, in which the model always begins, and two final states: one with reward +100 for students achieving higher-than-median normalized learning gain and one with reward -100 for the remaining students, following the conventions established in prior research into reinforcement learning for tutorial dialogue [18, 19].

Using these formalizations, one state was assigned to each of the log entries collected during the sessions and transition probabilities were computed between them when a tutor made an intervention (*TutorMove*) and when a tutor did not make an intervention (*NoMove*) based on the transition frequencies observed in the data. Any states that occurred less than once per session on average were combined into a single *LowFrequency* state, following the convention of prior work [23]. There were four states fitting this description: (*Task*, *Farther*, *StudentDial*), (*StudentDial*, *Farther*, *StudentDial*), (*StudentDial*, *Farther*, *Task*), and (*StudentDial*, *Farther*, *TutorDial*). Thus, the final MDP model contained 25 states requiring a tutorial intervention decision (23 states composed of feature combinations, the *LowFrequency* state, and the *Initial* state), and two final states.

The *Current Student Action* and *Last Action* features were relatively straightforward to assign to log entries by simply observing what a student was currently doing at that point in the session and observing what action had occurred most recently. The *Task Trajectory* feature was computed by discretizing the students' work on the task into chunks, which presents a substantial research question and design decision for supporting computer science learning. Historically, intelligent tutoring systems for computer science have utilized granularity at one extreme or the other. The smallest possible granularity is every keystroke, perhaps the earliest example of this being the Lisp tutor of Anderson and colleagues [24]. The largest granularity could arguably be to evaluate only when the student deems the artifact complete enough to manually submit for evaluation, which was the approach taken by another very early computer science tutor, Proust [25]. For the JavaTutor system, evaluating the student program more often than at the completion of tasks is essential to support dialogue, but an every-keystroke evaluation is too frequent due in part to algorithm runtime limitations. We define our task events as beginning when a student begins typing in the task pane and ending when a student has not typed in the task pane for at least 1.5 seconds. This threshold of 1.5 seconds was chosen empirically before model building to strike a balance between shorter thresholds, which resulted in frequent switching between "working on task" and "not working on task" states, and longer thresholds, which resulted in never leaving the "working on task" state.

After each task event (discretized as described above), a student's program was separated into tokens as defined by the Java compiler, and a token-level minimum edit distance was computed from that student's final solution for the lesson, tokenized in the same manner. Variable names, comments, and the contents of string literals were ignored in this edit distance calculation. The change in the edit distance from one chunk to the next determined the value of the *Task Trajectory* feature. Because the tutors were experienced in Java programming and had knowledge of the lesson structure, it is reasonable to assume that they were able to determine whether the student was moving farther or closer to the final solution. In this way, the edit distance algorithm provides a rough, automatically computable estimate of the tutors' assessment of student progress.

4 Policy Learning

The goal of this analysis is to learn a tutorial intervention policy—*when* to intervene—that reflects the most effective strategies within the corpus. In the MDP framework described above, this involves maximizing the learning gain reward. In order to learn this tutorial intervention policy, we used a policy iteration algorithm [21] on the MDP. For each iteration, this algorithm computes the expected reward in each state $s \epsilon S$ when taking each action $a \epsilon A$, based on the computed transition probabilities to other states and the expected rewards of those states from the previous iteration. Following the practice of prior work [13, 17], a discount factor of 0.9 was used to penalize delayed rewards (those requiring several state transitions to achieve) in favor of immediate rewards (those requiring few state transitions to achieve). The policy iteration continues until convergence is reached; that is, until the change in expected reward for each state is less than some epsilon value between iterations. We used an epsilon of 10^{-7}, requiring 125 iterations to converge. The resulting policy is shown in Table 2.

Table 2. The learned tutorial intervention policy

State (*Current Action*, *Task Trajectory*, *Last Action*)	**Policy**
(Task, Closer, Task)	TutorMove
(Task, Closer, StudentDial)	TutorMove
(Task, Closer, TutorDial)	TutorMove
(Task, Farther, Task)	TutorMove
(Task, Farther, TutorDial)	TutorMove
(Task, NoChange, Task)	TutorMove
(Task, NoChange, StudentDial)	NoMove
(Task, NoChange, TutorDial)	TutorMove
(StudentDial, Closer, Task)	TutorMove
(StudentDial, Closer, StudentDial)	TutorMove
(StudentDial, Closer, TutorDial)	TutorMove
(StudentDial, NoChange, Task)	NoMove
(StudentDial, NoChange, StudentDial)	NoMove
(StudentDial, NoChange, TutorDial)	NoMove
(NoAction, Closer, Task)	TutorMove
(NoAction, Closer, StudentDial)	TutorMove
(NoAction, Closer, TutorDial)	NoMove
(NoAction, Farther, Task)	NoMove
(NoAction, Farther, StudentDial)	TutorMove
(NoAction, Farther, TutorDial)	NoMove
(NoAction, NoChange, Task)	TutorMove
(NoAction, NoChange, StudentDial)	NoMove
(NoAction, NoChange, TutorDial)	NoMove
Initial	TutorMove
LowFrequency	TutorMove

Some noteworthy patterns emerge in the intervention policy learned from the corpus. For example, in seven of the eight states where the student is actively engaged in task actions (*Task*, *, *), the policy recommends that the tutor make a dialogue move. An excerpt from the corpus illustrating this strategy in a high learning gain session is shown in Figure 2, on lines 2-4. An excerpt from a low learning gain session showing tutor non-intervention during task progress is shown in Figure 3. In addition, among the states in which no action is currently being taken by the student and the last action was a tutor message, i.e., matching the pattern (*NoAction*, *, *TutorDial*), we find that the policy recommends that a tutor not make another consecutive dialogue move, regardless of how well the student is progressing on the task. However, Figure 2 shows that high learning gains are possible without strictly following this particular recommendation. Additional discussion on these recommendations can be found in [26].

5 Conclusion and Future Work

Current tutorial dialogue systems are highly effective, and matching the effectiveness of the most effective tutors is a driving force of tutorial dialogue research. This paper presents a step toward rich, adaptive dialogue for supporting computer science learning by introducing a representation of task-oriented dialogue with unrestricted turn-taking in a reinforcement learning framework and presenting initial results of an automatically learned policy for when to intervene. The presented approach will inform the development of the JavaTutor tutorial dialogue system, whose initial policies will be learned based on the fixed human-human corpus described here.

Event	Tutor action and state transition
1. *Student is declaring a String variable named "aStringVariable".*	*NoMove* ↓ (Task, NoChange, Task)
2. *Tutor starts typing a message*	*TutorMove*
3. *1.5 seconds elapse, task action is complete.*	↓
4. **Tutor message:** That works, but let's give the variable a more descriptive name	(NoAction, Closer, TutorDial)
5. *Tutor starts typing a message*	*TutorMove*
6. *Student starts typing a message*	↓
7. **Student message:** ok	
8. **Tutor message:** Usually, the variable's name tells us what data it has stored	(NoAction, Closer, TutorDial)

Fig. 2. An excerpt from a high learning gain session.

Event	Tutor action and state transition
1. *Student has just attempted to implement the programming code needed to complete the task, with no tutor intervention.*	*NoMove* ↓ (NoAction, Closer, Task)
2. *Student starts typing a message*	*NoMove* ↓ (StudentDial, Closer, Task)
3. **Student message:** not sure if this is right…	*NoMove* ↓ (NoAction, Closer, StudentDial)

Fig. 3. An excerpt from a low learning gain session.

Further exploring of the state space via simulation and utilizing a more expressive representation of state are highly promising directions for future work. Other directions for future work include undertaking a more fine-grained analysis of the timing of interventions, which could inform the development of more natural interactions, as well as allowing for more nuanced intervention strategies. Additionally, these models should be enhanced with a more expressive representation of both dialogue and task. It is hoped that these lines of investigation will yield highly effective machine-learned policies for tutorial dialogue systems and that tutorial dialogue systems for computer science will make this subject more accessible to students of all grade levels.

Acknowledgements

The JavaTutor project team includes Eric Wiebe, Bradford Mott, Eun Young Ha, Joseph Grafsgaard, Alok Baikadi, Megan Hardy, Mary Luong, Miles Smaxwell, Natalie Kerby, Robert Fulton, Caitlin Foster, Joseph Wiggins, and Denae Ford. This work is supported in part by the National Science Foundation through Grants DRL-1007962 and CNS-1042468. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the participants, and do not necessarily represent the official views, opinions, or policy of the National Science Foundation.

References

1. Bloom, B.: The 2 sigma problem: the search for methods of group instruction as effective as one-to-one tutoring. Educational Researcher. 13, 4–16 (1984).
2. VanLehn, K., Graesser, A.C., Jackson, G.T., Jordan, P., Olney, A., Rosé, C.P.: When are tutorial dialogues more effective than reading? Cognitive Science. 31, 3–62 (2007).
3. Evens, M.W., Michael, J.: One-on-One Tutoring by Humans and Computers. Lawrence Erlbaum Associates, Mahwah, New Jersey (2005).
4. Heffernan, N.T., Koedinger, K.: The design and formative analysis of a dialog-based tutor. Workshop on Tutorial Dialogue Systems. pp. 23–34 (2001).
5. Forbes-Riley, K., Litman, D.: Adapting to student uncertainty improves tutoring dialogues. Proceedings of the International Conference on Artificial Intelligence in Education. pp. 33–40 (2009).
6. Kersey, C., Di Eugenio, B., Jordan, P., Katz, S.: KSC-PaL: A peer learning agent that encourages students to take the initiative. Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications. pp. 55–63 (2009).
7. Kumar, R., Rosé, C.P.: Architecture for Building Conversational Agents that Support Collaborative Learning. IEEE Transactions on Learning. 4, 21–34 (2011).
8. Jackson, G.T., Person, N.K., Graesser, A.C.: Adaptive Tutorial Dialogue in AutoTutor. ITS 2004 Workshop Proceedings on Dialog-based Intelligent Tutoring Systems. pp. 9–13 (2004).
9. D'Mello, S., Graesser, A.: AutoTutor and affective autotutor: Learning by talking with cognitively and emotionally intelligent computers that talk back. ACM Transactions on Interactive Intelligent Systems. 2, (2012).
10. Jonsdottir, G.R., Thorisson, K.R., Nivel, E.: Learning Smooth, Human-Like Turntaking in Realtime Dialogue. Proceedings of the 8th International Conference on Intelligent Virtual Agents. pp. 162–175 (2008).
11. Ward, N.G., Fuentes, O., Vega, A.: Dialog Prediction for a General Model of Turn-Taking. Proceedings of the International Conference on Spoken Language Processing (2010).

12. Raux, A., Eskenazi, M.: Optimizing the turn-taking behavior of task-oriented spoken dialog systems. ACM Transactions on Speech and Language Processing. 9, 1–23 (2012).
13. Bohus, D., Horvitz, E.: Multiparty Turn Taking in Situated Dialog: Study, Lessons, and Directions. Proceedings of the 12th Annual Meeting of the Special Interest Group in Discourse and Dialogue. pp. 98–109 (2011).
14. Morbini, F., Forbell, E., DeVault, D., Sagae, K., Traum, D.R., Rizzo, A.A.: A Mixed-Initiative Conversational Dialogue System for Healthcare. Proceedings of the 13th Annual Meeting of the Special Interest Group in Discourse and Dialogue. pp. 137–139 (2012).
15. Mitchell, C.M., Boyer, K.E., Lester, J.C.: From strangers to partners: examining convergence within a longitudinal study of task-oriented dialogue. Proceedings of the 13th Annual SIGDIAL Meeting on Discourse and Dialogue. pp. 94–98 (2012).
16. Ha, E.Y., Grafsgaard, J.F., Mitchell, C.M., Boyer, K.E., Lester, J.C.: Combining verbal and nonverbal features to overcome the "information gap" in task-oriented dialogue. Proceedings of the 13th Annual SIGDIAL Meeting on Discourse and Dialogue. pp. 246–256 (2012).
17. Grafsgaard, J.F., Fulton, R., Boyer, K.E., Wiebe, E., Lester, J.C.: Multimodal analysis of the implicit affective channel in computer-mediated textual communication. to appear in Proceedings of the 14th ACM international conference on Multimodal Interaction (2012).
18. Tetreault, J.R., Litman, D.J.: A Reinforcement Learning approach to evaluating state representations in spoken dialogue systems. Speech Communication. 50, 683–696 (2008).
19. Chi, M., VanLehn, K., Litman, D.: Do micro-level tutorial decisions matter: applying reinforcement learning to induce pedagogical tutorial tactics. Proceedings of the International Conference on Intelligent Tutoring Systems. pp. 224–234. (2010).
20. Marx, J.D., Cummings, K.: Normalized change. American Journal of Physics. 75, 87–91 (2007).
21. Sutton, R., Barto, A.: Reinforcement Learning. MIT Press, Cambridge, MA (1998).
22. Singh, S., Litman, D., Kearns, M., Walker, M.: Optimizing Dialogue Management with Reinforcement Learning: Experiments with the NJFun System. Journal of Artificial Intelligence Research. 16, 105–133 (2002).
23. Tetreault, J.R., Litman, D.J.: Using Reinforcement Learning to Build a Better Model of Dialogue State. Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics. pp. 289–296 (2006).
24. Anderson, J.R., Boyle, C.F., Corbett, A.T., Lewis, M.W.: Cognitive modeling and intelligent tutoring. Artificial Intelligence. 42, 7–49 (1990).
25. Johnson, W.L., Soloway, E.: PROUST: Knowledge-based program understanding. ICSE '84: Proceedings of the 7th international conference on Software engineering. pp. 369–380 (1984).
26. Mitchell, C.M., Boyer, K.E., Lester, J.C.: A Markov Decision Process Model of Tutorial Intervention in Task-Oriented Dialogue. To appear in Proceedings of the 16th International Conference on Artificial Intelligence in Education (2013).

Automatic Generation of Programming Feedback: A Data-Driven Approach

Kelly Rivers and Kenneth R. Koedinger

Carnegie Mellon University

Abstract. Automatically generated feedback could improve the learning gains of novice programmers, especially for students who are in large classes where instructor time is limited. We propose a data-driven approach for automatic feedback generation which utilizes the program solution space to predict where a student is located within the set of many possible learning progressions and what their next steps should be. This paper describes the work we have done in implementing this approach and the challenges which arise when supporting ill-defined domains.

Keywords: automatic feedback generation; solution space; computer science education; intelligent tutoring systems

1 Introduction

In the field of learning science, feedback is known to be important in the process of helping students learn. In some cases, it is enough to tell a student whether they are right or wrong; in others, it is better to give more details on why a solution is incorrect, to guide the student towards fixing it. The latter approach may be especially effective for problems where the solution is complex, as it can be used to target specific problematic portions of the student's solution instead of throwing the entire attempt out. However, it is also more difficult and time-consuming to provide.

In computer science education, we have been able to give students a basic level of feedback on their programming assignments for a long time. At the most basic level, students can see whether their syntax is correct based on feedback from the compiler. Many teachers also provide automated assessment with their assignments, which gives the student more semantic information on whether or not their attempt successfully solved the problem. However, this feedback is limited; compiler messages are notoriously unhelpful, and automated assessment is usually grounded in test cases, which provide a black and white view of whether the student has succeeded. The burden falls on the instructors and teaching assistants (TAs) to explain to students why their program is failing, both in office hours and in grading. Unfortunately, instructor and TA time is limited, and it becomes nearly impossible to provide useful feedback when course sizes become larger and massive open online courses grow more common.

Given this situation, a helpful approach would be to develop a method for automatically generating more content-based and targeted feedback. An automatic approach could scale easily to large class sizes, and would hopefully be able to handle a large portion of the situations in which students get stuck. This would greatly reduce instructor grading time, letting them focus on the students who struggle the most. Such an approach is easier to hypothesize than it is to create, since student solutions are incredibly varied in both style and algorithmic approach and programming problems can become quite complex. An automatic feedback generation system would require knowledge of how far the student had progressed in solving the problem, what precisely was wrong with their current solution, and what constraints were required in the final, correct solutions.

In this paper, we propose a method for creating this automatic feedback by utilizing the information made available by large corpuses of previous student work. This data can tell us what the most common correct solutions are, which misconceptions normally occur, and which paths students most often take when fixing their bugs. As the approach is data-driven, it requires very little problem-specific input from the teacher, which makes it easily scalable and adaptable. We have made significant progress in implementing this approach and plan to soon begin testing it with real students in the field.

2 Solution Space Representation

Our method relies upon the use of *solution spaces*. A solution space is a graph representation of all the possible paths a student could take in order to get from the problem statement to a correct answer, where the nodes are candidate solutions and the edges are the actions used to move from one solution state to another. Solution spaces can be built up from student data by extracting students' learning progressions from their work and inserting them into the graph as a directed chain. Identical solutions can be combined, which will represent places where a student has multiple choices for the next step to take, each of which has a different likelihood of getting them to the next answer.

A solution space can technically become infinitely large (especially when one considers paths which do not lead to a correct solution), but in practice there are common paths which we expect the student to take. These include the learning progression that the problem creator originally intended, other progressions that instructors and teaching assistants favor, and paths that include any common misconceptions which instructors may have recognized in previous classes. If we can recognize when a student is on a common path (or recognize when the student has left the pack entirely) we can give them more targeted feedback on their work.

While considering the students' learning progressions, we need to decide at what level of granularity they should be created. We might consider very small deltas (character or token changes), or very large ones (save/compile points or submissions), depending on our needs. In our work we use larger deltas in order to examine the points at which students deliberately move from one state to the next; every time a student saves, they are pausing in their stream of work and often checking to see what changes occur in their program's output. Of course, this approach cannot fully represent all of the work that a student does; we cannot see the writing they are doing offline or hear them talking out ideas with their TAs. These interactions will need to be inferred from the changes in the programs that the student writes if we decide to account for them.

It is simple to create a basic solution space, but making the space *usable* is a much more difficult task. Students use different variable names, indentations, and styles, and there are multitudes of ways for them to solve the same problem with the same general approach. In fact, we do not want to see two different students submitting exactly the same code– if they do, we might suspect them of cheating! But the solution space is of no use to us if we cannot locate new students inside of it. Therefore, we need to reduce the size of the solution space by combining all semantically equivalent program states into single nodes.

Many techniques have been developed already for reducing the size of the solution spaces of ill-defined problems. Some represent the solution states with sets of constraints [5], some use graph representations to strip away excess material [4], and others use transformations to simplify solution states [9, 8]. We subscribe to the third approach by transforming student programs into *canonical forms* with a set of normalizing program transformations. These transformations simplify, anonymize, and order the program's syntax without changing its semantics, mapping each individual program to a normalized version. All transformations are run over abstract syntax trees (ASTs), which are parse trees for programs that remove extraneous whitespace and comments. If two different programs map to the same canonical form, we know that they must be semantically equivalent, so this approach can safely be used to reduce the size of the solution space.

Example Let us consider a very simple programming problem from one of the first assignments of an introductory programming class. The program takes as input an integer between 0 and 51, where each integer maps to a playing card, and asks the student to return a string representation of that card. The four of diamonds would map to 15, as clubs come before diamonds; therefore, given an input of 15, the student would return "4D". This problem tests students' ability to use mod and div operators, as well as string indexing. One student's incorrect solution to this problem is shown in Figure 1.

```
def intToPlayingCard(value):
    faceValue = value%13
    #use remainder as an index to get the face
    face = "23456789YJQKA"[faceValue]
    suitValue = (value-faceValue)%4
    suit = "CDHS"[suitValue]
    return face+suit
```

Fig. 1. A student's attempt to solve the playing card problem.

To normalize this student's program, we first extract the abstract syntax tree from the student's code, as is partially shown in Figure 2. All of the student's variables are immediately anonymized; in this case, 'value' will become 'v0', 'faceValue' will be 'v1', etc. We then run all of our normalizing transformations over the program, to see which ones will have an effect; in this case, the only transformation used is *copy propagation.* This transformation reduces the list of five statements in the program's body to a single return statement by copying the value assigned to each variable into the place where the variable is used later on. Part of the resulting canonical form is displayed in Figure 2. The new tree is much smaller, but the program will have the same effect.

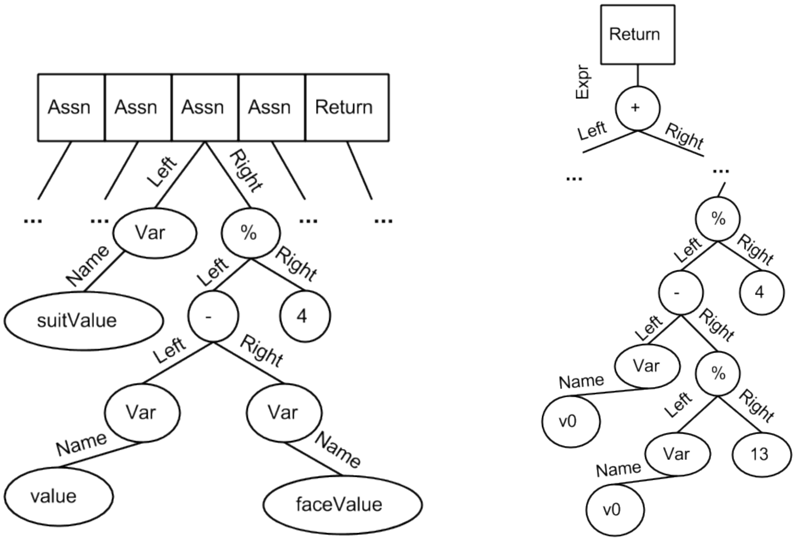


Fig. 2. Subparts of the student's ASTs, before (left) and after (right) normalization.

We have already implemented this method of solution space reduction and tested it with a dataset of final submissions from a collection of introductory programming problems. The method is quite effective, with the solution space size being reduced by slightly over 50% for the average problem [7]. However, we still find a long tail of singleton canonical forms existing in each problem's solution space, usually due to students who found strange, unexpected approaches or made unconventional mistakes. This long tail of unusual solutions adds another layer of complexity to the problem, as it decreases the likelihood that a new student solution will appear in the old solution space.

Our work so far has concentrated only on final student submissions, not on the paths students take while solving their problems. This could be seen as problematic, as we are not considering the different iterations a student might go through while working. However, our very early analysis of student learning progressions from a small dataset has indicated that students are not inclined to use incremental approaches. The students we observed wrote entire programs in single sittings, then debugged until they could get their code to perform correctly.

This suggests that our work using final program states may be close enough to the real, path-based solution space to successfully emulate it.

We note that the solution space is easiest to traverse and create when used on simple problems; as the required programs become longer, the number of individual states in the space drastically increases. We believe it may be possible to address this situation by breaking up larger problems into hierarchies of subproblems, each of which may map to a specific chunk of code. Then each subproblem can have its own solution space that may be examined separately from the other subproblems, and feedback can be assigned for each subproblem separately.

3 Feedback Generation

Once the solution space has been created, we need to consider how to generate feedback with it. The approach we have adopted is based on the Hint Factory [1], a logic tutor which uses prior data to give stuck students feedback on how to proceed. In the Hint Factory, each node of the solution space was the current state of the student's proof, and each edge was the next theorem to apply that would help the student move closer to the complete proof. The program used a Markov Decision Process to decide which next state to direct the student towards, optimizing for the fastest path to the solution.

Our approach borrows heavily from the Hint Factory, but also expands it. This is due to the ill-defined nature of solving programming problems, which specifies that different solutions can solve the same problem; this complicates several of the steps used in the original logic tutor. In this section we highlight three challenges that need to be addressed in applying the Hint Factory methodology to the domain of programming, and describe how to overcome each of them.

Other attempts have been made at automatic generation of feedback, both in the domain of programming and in more domain-general contexts. Some feedback methods rely on domain knowledge to create messages; Paquette et al.'s work on supporting domain-general feedback is an example of this [6]. Other methods rely instead on representative solutions, comparing the student's solution to the expected version. Examples here include Gerdes et al.'s related work on creating functional tutoring systems (which use instructor-submitted representative solutions) [2] and Gross et al.'s studies on using clustering to provide feedback (which, like our work, use correct student solutions) [3]. Though our work certainly draws on many of the elements used in these approaches, we explore the problem from a different angle in attempting to find entire paths to the closest solution (which might involve multiple steps), rather than jumping straight from the student's current state to the final solution. Whether this proves beneficial will remain to be seen in future studies.

3.1 Ordering of Program States

Our first challenge relates to the process of actually mapping out the suggested learning progressions for the student. Even after reducing the size of the solution space, there are still a large number of distinct solutions which are close yet not connected by previously-found learning paths. These close states can be helpful, as they provide more opportunities for students to switch between different paths while trying to reach the solution. Therefore, we need to connect each state to those closest to it, then determine which neighboring state will set the student on the best path to get to a final solution.

One obvious method for determining whether two states are close to each other would involve using tree edit distance, to determine how many changes needed to be made. However, this metric does not seem to work particularly well in practice; the weight of an edit is difficult to define, which makes comparing edits non-trivial. Instead, we propose the use of string edit distance (in this case, Levenshtein distance) to determine whether two programs are close to each other. To normalize the distances between states, we calculate the percentage similarity with $(l - distance)/l$ (where l is the length of the longer program); this ensures that shorter programs do not have an advantage over longer ones and results from different problems can easily be compared to each other. Once the distances between all programs have been calculated, a cut-off point can be determined that will separate close state pairs from far state pairs. Our early experimentation with this method shows that it is efficient on simple programs and produces pairs of close states for which we can generate artificial actions.

Once the solution space has been completely generated and connected, we need to consider how to find the best path from state A to state B. The algorithm for finding this will be naturally recursive in nature– the best path from A to B will be the best element of the set of paths S, where S is composed of paths from each neighbor of A to B. Paths which require fewer intermediate steps will be preferred, as they require the student to make less changes, but we also need to consider the distances between the program states. We can again use string edit distance to find these distances, or we can use the tree edits to look at the total number of individual changes required. Finally, we can use test cases to assign correctness parameters to each program state (as there are certainly some programs which are more incorrect than others); paths which gradually increase the number of test cases that a student passes may be considered more beneficial than paths which jump back and forth, as the latter paths may lead to discouragement and frustration in students.

Example In the previous section, we had found the canonical form for the student's solution; that form was labeled #22 in the set of all forms. As we were using a dataset of final submissions, we had no learning progressions to work with, we computed the normalized Levenshtein distance between each pair of states and connected those which had a percentage similarity of 90% or higher, thus creating a progression graph.

In Figure 3, we see that state #22 was connected to three possible next states: #4, #34, and #37. We know that #34 is incorrect, so it does not seem like a good choice; on the other hand, #4 and #37 are equally close to #22 and are both correct. State #37 had been reached by thirty students, while state #4 had only been reached by four; since #37 is more commonly used, it is probably the better target solution for the student.

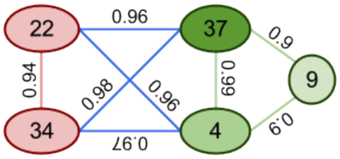


Fig. 3. The program state graph surrounding state #22. Red nodes are incorrect, green correct; a darker node indicates that more students used that approach.

3.2 Generating content deltas between states

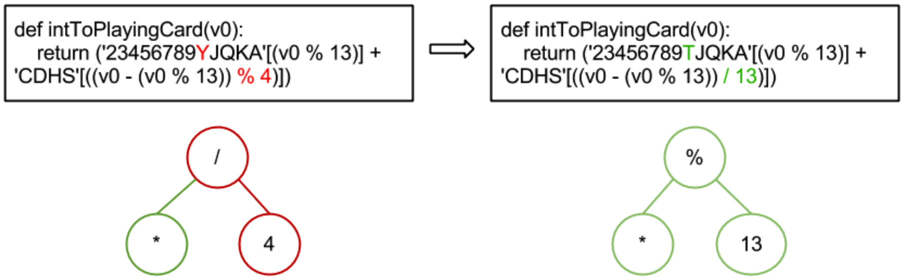
Next, we face the challenge of determining how to extract the content of the feedback message from the solution space. The feedback that we give the student comes from the edge between the current and target states, where that edge represents the actions required to get from one state to the other. In well-defined domains, these actions are often simple and concrete, but they become more complex when the problems are less strictly specified.

Before, we used string distance to determine how similar two programs were, in order to find distances quickly and easily. Now that we need to know what the differences actually are, we use tree edits to find the additions, deletions, and changes required to turn one tree into another. It is moderately easy to compute these when comparing ordinary trees, but ASTs add an extra layer of complexity as there are certain nodes that hold lists of children (for example, the bodies of loops and conditionals), where one list can hold more nodes than another. To compare these nodes, we find the maximal ordered subset of children which appear in both lists; the leftover nodes can be considered changes.

After we have computed these edits, we can use them to generate feedback for the student in the traditional way. Cognitive tutors usually provide three levels of hints; we can use the same approach here, first providing the location of an error, then the token which is erroneous, and finally what the token needs to be changed to in order to fix the error. In cases where more than one edit needs to be made the edits can be provided to the student one at a time, so that the student has a chance to discover some of the problems on their own.

It may be possible to map certain edit patterns to higher-level feedback messages, giving students more conceptual feedback. Certain misconceptions and mistakes commonly appear in novice programs; accidental use of integer division and early returns inside of loops are two examples. If we can code the patterns that these errors commonly take (in these cases, division involving two integer values and return statements occurring in the last line of a loop's body), we can provide higher-level static feedback messages that can be provided to students instead of telling them which values to change. This may help them recognize such common errors on their own in future tasks.

Example To generate the feedback message in our continuing example, we find the tree edits required to get from state #22 to state #37. These come in two parts: one a simple change, the other a more complex edit. Both are displayed in Figure 4. The first change is due to a typo in the string of card face values that the student is indexing (Y instead of T for ten); as the error occurs in a leaf node (a constant value), pointing it out and recommending a change is trivial. Such a feedback message might look like this: *In the return statement, the string ”23456789YJQKA” should be ”23456789TJQKA”.*



```
def intToPlayingCard(v0):
   return ('23456789YJQKA'[(v0 % 13)] +
'CDHS'[((v0 - (v0 % 13)) % 4)])
```

```
def intToPlayingCard(v0):
   return ('23456789TJQKA'[(v0 % 13)] +
'CDHS'[((v0 - (v0 % 13)) / 13)])
```

Fig. 4. The change found between the two programs, represented in text and tree format (with * representing a further subexpression).

The second error is due to a misconception about how to find the index of the correct suit value. In the problem statement, the integer card values mapped cards first by face value and then by suit; all integers from 0 to 12 would be clubs, 13 to 23 would be diamonds, etc. This is a step function, so the student should have used integer division to get the correct value. In a terrible twist of fate, this part of the student's code will actually work properly; $v0 - (v0\%13)$ returns the multiple of 13 portion of v0, and the first four multiples of 13 (0, 13, 26, and 39) each return the correct index value when modded by 4 (0, 1, 2, and 3). Still, it seems clear that the student is suffering from a missing piece of knowledge, as it would be much simpler to use the div operator.

In the AST, the two solutions match until they reach the value used by the string index node. At that point, one solution will use mod while the other uses div, and one uses a right operand of 4 while the other uses a right operand of 13. It's worth noting, however, that both use the same subexpression in the left operand; therefore, in creating feedback for the student, we can leave that part out. Here, the feedback message might be this: *In the right side of the addition in the return statement, use div instead of mod.* The further feedback on changing 4 to 13 could be provided if the student needed help again later.

3.3 Reversing deltas to regain content

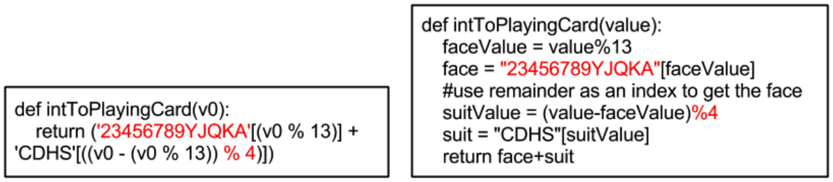
Finally, we need to take the content of the feedback message which we created in the previous part and map it back to the student's original solution. If the student's solution was equivalent to the program state, this would be easy– however, because we had to normalize the student programs, we will need to map the program state back up to the individual student solution in order to create their personal feedback message.

In some situations, this will be easy. For example, it's possible that a student solution only had whitespace cleaned up and variable names anonymized; if this was the case, the location of the code would remain the same, and variable names could be changed in the feedback message easily. In many other cases, the only transformations applied would be ordering and propagation functions; for these, we can keep track of where each expression occurred in the original program, then map the code segments we care about back to their positions in the original code. Our running example falls into this "easy" category; even though the student's program looks very different from the canonical version, we only need to unroll the copy propagation to get the original positions back.

Other programs will present more difficulties. For example, any student program which has been reduced in size (perhaps through constant folding, or conditional simplification) might have a feedback expression which needs to be broken into individual pieces. One solution for this problem would be to record each transformation that is performed on a program, then backtrack through them when mapping feedback. Each transformation function can be paired with a corresponding "undo" function that will take the normalized program and a description of what was changed, then generate the original program.

Example All of the program transformations applied to the original student program in order to produce state #22 were copy propagations; each variable was copied down into each of its references and deleted, resulting in a single return statement. To undo the transformations, the expressions we want to give feedback on ('23456789YJQKA' and $(v0-(v0\%13))\%4$) must be mapped to the variables that replace them– face and suit (where suit is later mapped again to suitValue). We can then examine the variable assignment lines to find the original location in which the expression was used (see Figure 5), which maps the expressions to lines 2 and 4. The first expression's content is not modified, but the second changes into $(value - faceValue)\%4$. This change lies outside of the feedback that we are targeting, so it does not affect the message.

After the new locations have been found, the feedback messages are correspondingly updated by changing the location that the message refers to. In this case, the first feedback message would change to: In the **second line**, the string '23456789YJQKA' should be '23456789TJQKA'. The second would become: In the **fourth line**, use div instead of mod.



```
def intToPlayingCard(v0):
    return ('23456789YJQKA'[(v0 % 13)] +
'CDHS'[((v0 - (v0 % 13)) % 4)])
```

```
def intToPlayingCard(value):
    faceValue = value%13
    face = "23456789YJQKA"[faceValue]
    #use remainder as an index to get the face
    suitValue = (value-faceValue)%4
    suit = "CDHS"[suitValue]
    return face+suit
```

Fig. 5. A comparison of the canonical (left) and original (right) programs. The code snippets we need to give feedback on are highlighted.

4 Conclusion

The approach we have described utilizes the concept of solution spaces to determine where a new student is in their problem-solving process, then determines what feedback to provide by traversing the space to find the nearest correct solution. Representing the solution space has been implemented and tested, but generating feedback is still in progress; future work will determine how often it is possible to provide a student with truly useful and usable feedback.

Acknowledgements. This work was supported in part by Graduate Training Grant awarded to Carnegie Mellon University by the Department of Education (# R305B090023).

References

1. Barnes, Tiffany, and John Stamper. "Toward automatic hint generation for logic proof tutoring using historical student data." Intelligent Tutoring Systems. Springer Berlin Heidelberg, 2008.
2. Gerdes, Alex, Johan Jeuring, and Bastiaan Heeren. "An interactive functional programming tutor." Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education. ACM, 2012.
3. Gross, Sebastian, et al. "Cluster based feedback provision strategies in intelligent tutoring systems." Intelligent Tutoring Systems. Springer Berlin Heidelberg, 2012.
4. Jin, Wei, et al. "Program representation for automatic hint generation for a data-driven novice programming tutor." Intelligent Tutoring Systems. Springer Berlin Heidelberg, 2012.
5. Le, Nguyen-Thinh, and Wolfgang Menzel. "Using constraint-based modelling to describe the solution space of ill-defined problems in logic programming." Advances in Web Based LearningICWL 2007. Springer Berlin Heidelberg, 2008. 367-379.
6. Paquette, Luc, et al. "Automating next-step hints generation using ASTUS." Intelligent Tutoring Systems. Springer Berlin Heidelberg, 2012.
7. Rivers, Kelly, and Kenneth R. Koedinger. "A canonicalizing model for building programming tutors." Intelligent Tutoring Systems. Springer Berlin Heidelberg, 2012.
8. Weragama, Dinesha, and Jim Reye. "Design of a knowledge base to teach programming." Intelligent Tutoring Systems. Springer Berlin Heidelberg, 2012.
9. Xu, Songwen, and Yam San Chee. "Transformation-based diagnosis of student programs for programming tutoring systems." Software Engineering, IEEE Transactions on 29.4 (2003): 360-384.

JavaParser: A Fine-Grain Concept Indexing Tool for Java Problems

Roya Hosseini, Peter Brusilovsky

University of Pittsburgh, Pittsburgh, USA
{roh38,peterb}@pitt.edu

Abstract. Multi-concept nature of problems in the domain of programming languages requires fine-grained indexing which is critical for sequencing purposes. In this paper, we propose an approach for extracting this set of concepts in a reliable automated way using JavaParser tool. To demonstrate the importance of fine-grained sequencing, we provide an example showing how this information can be used for problem sequencing during exam preparation.

Keywords: indexing, sequencing, parser, java programming

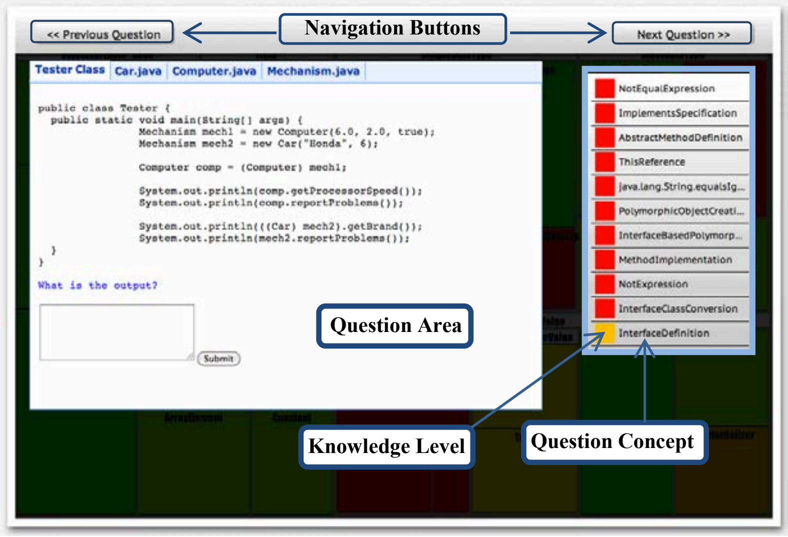
1 Introduction

One of the oldest functions performed by adaptive educational systems is guiding students to most appropriate educational problems at any time of their learning process. In classic ICAI and ITS system this function was known as task sequencing [1; 6]. In modern hypermedia-based systems it is more often referred as navigation support. The intelligent decision mechanism behind these approaches is typically based on a domain model that decomposes the domain into a set of knowledge units. This domain model serves as a basis of student overlay model and as a dictionary to index educational problems or tasks. Considering the learning goal and the current state of student knowledge reflected by the student model, various sequencing approaches are able to determine which task is currently the most appropriate.

An important aspect of this decision process is the granularity of the domain model and the related granularity of task indexing. In general, the finer are the elements of the domain model and the more precise is task indexing, the better precision could be potentially offered by the sequencing algorithm in determining the best task to solve. However, fine-grained domain models that dissect a domain into many dozens to many hundreds of knowledge units are much harder to develop and to use for indexing. As a result, many adaptive educational systems use relatively coarse-grained models where a knowledge unit corresponds to a considerably-sized topic of learning material, sometimes even a whole lecture.. With these coarse-grain models, each task is usually indexed with just 1-3 topics. In particular, this approach is used by the majority of adaptive systems in the area of programming [2; 4; 5; 7].

Our past experience with adaptive hypermedia systems for programming [2; 4] demonstrated that adaptive navigation support based on coarse grain problem indexing is surprisingly effective way to guide students over their coursework, yet it doesn’t work well in special cases such as remediation or exam preparation. In these special situations students might have a reasonable overall content understanding (i.e., coarse-grain student model registers good knowledge), while still possessing some knowledge gaps and misconceptions that could be only registered using a finer-grain student model.. In this situation only a fine-grain indexing and sequencing is able to suggest learning tasks that can address these gaps and misconceptions.

To demonstrate the importance of fine-grained indexing, we can look at an example of a system called *Knowledge Maximizer* [3] that uses fine-grain concept-level problem indexing to identify gaps in user knowledge for exam preparation. This system assumes a student already did considerable amount of work and the goal is to help her define gaps in knowledge and try to fix that holes as soon as possible. Fig. 1 represents the Knowledge Maximizer interface. The question with the highest rank is shown first. User can navigate the ranked list of questions using navigation buttons at the top. Right side of the panel shows the list of fine-grained concepts covered by the question. The color next to each concept visualizes the student's current knowledge level (from red to green). Evaluation results confirm that using fine-grained indexing in Knowledge Maximizer has positive effect on students' performance and also shorten the time for exam preparation.



```
public class Tester {
  public static void main(String[] args) {
                Mechanism mech1 = new Computer(6.0, 2.0, true);
                Mechanism mech2 = new Car("Honda", 6);

                Computer comp = (Computer) mech1;

                System.out.println(comp.getProcessorSpeed());
                System.out.println(comp.reportProblems());

                System.out.println(((Car) mech2).getBrand());
                System.out.println(mech2.reportProblems());
  }
}
```

What is the output?

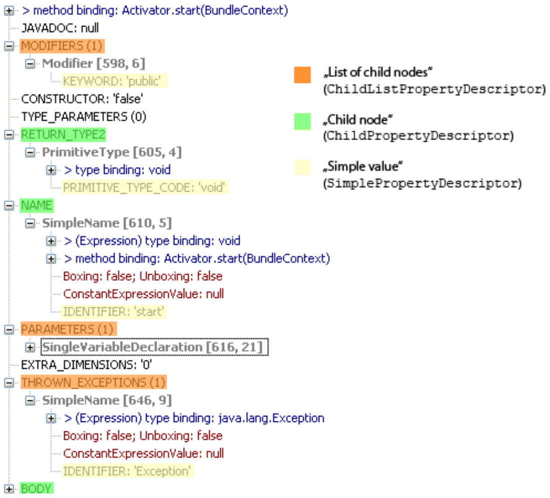
Fig. 1. The Knowledge Maximizer interface.

The problem with finer-grain indexing, such as used by the Knowledge Maximizer is the high cost of indexing. While fine-grain domain model has to be developed just once, the indexing process has to be repeated for any new question. Given that most complex questions used by the system include over 90 concepts each, the high cost of indexing effectively prevents an expansion of the body of problems. To resolve this problem, we developed an automatic approach for fine-grained indexing for programming problems in Java based on program parsing. This approach is presented in the following section.

2 Java Parser

Java parser is a tool that we developed to index Java programs with concepts of Java ontology developed by our group (http://www.sis.pitt.edu/~paws/ont/java.owl). This tool provides the user with semi-automated indexing support during developing new learning materials for the Java Programming Language course. This parser is developed using the Eclipse Abstract Syntax Tree framework. This framework generates an Abstract Syntax Tree (AST) that entirely represents the program source. AST consists of several nodes each containing some information known as *structural properties*. For example, Fig. 2 shows structural properties for the following method declaration:

```
public void start(BundleContext context) throws Exception {
        super.start(context);
}
```



```
⊞ > method binding: Activator.start(BundleContext)
  JAVADOC: null
⊟ MODIFIERS (1)
    ⊟ Modifier [598, 6]
        KEYWORD: 'public'
  CONSTRUCTOR: 'false'
  TYPE_PARAMETERS (0)
⊟ RETURN_TYPE2
    ⊟ PrimitiveType [605, 4]
        ⊞ > type binding: void
        PRIMITIVE_TYPE_CODE: 'void'
⊟ NAME
    ⊟ SimpleName [610, 5]
        ⊞ > (Expression) type binding: void
        ⊞ > method binding: Activator.start(BundleContext)
        Boxing: false; Unboxing: false
        ConstantExpressionValue: null
        IDENTIFIER: 'start'
⊟ PARAMETERS (1)
    ⊞ SingleVariableDeclaration [616, 21]
  EXTRA_DIMENSIONS: '0'
⊟ THROWN_EXCEPTIONS (1)
    ⊟ SimpleName [646, 9]
        ⊞ > (Expression) type binding: java.lang.Exception
        Boxing: false; Unboxing: false
        ConstantExpressionValue: null
        IDENTIFIER: 'Exception'
⊞ BODY
```

🟧 „List of child nodes“ (ChildListPropertyDescriptor)

🟩 „Child node“ (ChildPropertyDescriptor)

▢ „Simple value“ (SimplePropertyDescriptor)

Fig. 2. Structural properties of a method declaration

Table 1. Sample of JavaParser output

Source	Output

`public void`
`start(BundleContext context)`
`throws Exception {`
`        super.start(context);`
`}`
	Super Method Invocation,
	Public Method Declaration,
	Exception,
	Formal Method Parameter,
	Single Variable Declaration,
	Void

After building the tree using *Eclipse AST API*, the parser performs a semantic analyzed using the information in each node. This information is used to identify fine-grained indexes for the source program. Table 1 shows the output concepts of *JavaParser* for the code fragment mentioned above. Note that the goal of the parser is to detect the lowest level ontology concepts behind the code since the upper level concepts can be deduced using ontology link propagation. For example, as you see in Table 1, parser detects “void” and “main” ignoring upper-level concept of “modifier”.

We compared the accuracy of *JavaParser* with manual indexing for 103 Java problems and found out that our parser was able to index 93% of the manually indexed concepts. Therefore, automatic parser can replace time-consuming process of manual indexing with a high precision and open the way to community-driven problem authoring and targeted expansion of the body of problems.

3 Conclusion

Having fine-grained indexing for programming problems is necessary for better sequencing of learning materials for students; however, the cost of manual fine-grained indexing is prohibitively high. In this paper, we presented a fine grained indexing approach and tool for automatic indexing of Java problems. We also showed an application of fine-grained problem indexing during exam preparation where small size of knowledge units is critical for finding sequence of problems that fills the gaps in student knowledge. Results show that proposed automatic indexing tool can offer the quality of indexing that is comparable with manual indexing by expert for a fraction of its cost.

References

1. Brusilovsky, P.: A framework for intelligent knowledge sequencing and task sequencing. In: Proc. of Second International Conference on Intelligent Tutoring Systems, ITS'92. Springer-Verlag (1992) 499-506
2. Brusilovsky, P., Sosnovsky, S., Yudelson, M.: Addictive links: The motivational value of adaptive link annotation. New Review of Hypermedia and Multimedia 15, 1 (2009) 97-118
3. Hosseini, R., Brusilovsky, P., Guerra, J.: Knowledge Maximizer: Concept-based Adaptive Problem Sequencing for Exam Preparation. In: Proc. of the 16th International Conference on Artificial Intelligence in Education. (2013) In Press
4. Hsiao, I.-H., Sosnovsky, S., Brusilovsky, P.: Guiding students to the right questions: adaptive navigation support in an E-Learning system for Java programming. Journal of Computer Assisted Learning 26, 4 (2010) 270-283
5. Kavcic, A.: Fuzzy User Modeling for Adaptation in Educational Hypermedia. IEEE Transactions on Systems, Man, and Cybernetics 34, 4 (2004) 439-449
6. McArthur, D., Stasz, C., Hotta, J., Peter, O., Burdorf, C.: Skill-oriented task sequencing in an intelligent tutor for basic algebra. Instructional Science 17, 4 (1988) 281-307
7. Vesin, B., Ivanović, M., Klašnja-Milićević A., Budimac, Z.: Protus 2.0: Ontology-based semantic recommendation in programming tutoring system. Expert Systems with Applications 39, 15 (2012) 12229-12246

AIED 2013 Workshops Proceedings
Volume 10

Self-Regulated Learning in Educational Technologies: Supporting, modeling, evaluating, and fostering metacognition with computer-based learning environments (SRL@ET)

Workshop Co-Chairs:

Amali Weerasinghe
ICTG, Department of Computer Science and Software Engineering,
University of Canterbury, NZ

Benedict du Boulay
HCT, Department of Informatics, University of Sussex, UK

Gautam Biswas
School of Engineering, Vanderbilt University, USA

http://workshops.shareghi.com/AIED2013/

Preface

It is important that the educational system helps learners develop a general ability to get up to speed quickly in new domains. In order to do that students need to be able to manage their learning, for example, by setting goals, planning their learning, monitoring their progress, and responding appropriately to difficulties and errors. These general learning skills are often referred to as metacognition, or self-regulated learning (SRL). Bransford et al. [3] suggest focusing on metacognition as one of three principles that should be applied to educational research and design, as stated in the influential volume “How People Learn.” A similar recommendation is given also in Clark and Mayer's [4] book about e-learning design principles. Azevedo and colleagues have found that students who regulate their learning in a hypermedia environment are more likely to acquire deep understanding of the target domain [2]. A key question is whether instructional technology can be as effective in fostering metacognitive skills as it is in teaching domain-specific skills and knowledge. Numerous learning environments include metacognitive support in order to improve domain-level learning (e.g., [5] and [1] support self-explanation in order to promote learning of Physics and Geometry, respectively.) However, only a few systems actually attempt to help students to acquire or improve the metacognitive skills themselves (and not only the domain-level knowledge). Some work suggests that improving metacognitive and SRL skills can be done using educational technologies. Examples include the Help Tutor [6], Betty’s Brain [7] and MetaTutor [2]. However, a lot remains to be known about the fashion in which educational technologies can support the acquisition of metacognitive and SRL skills. The modeling, tutoring, and evaluation of metacognitive skills and knowledge poses a number of challenges:

Modeling metacognitive and SRL knowledge: Metacognitive knowledge is ill-defined by nature. While the correct answer to a problem at the domain level is usually independent of the learner or the context, this is not the case for metacognitive dilemmas, in which the appropriate metacognitive actions depend on the student, her capabilities, motivation, preferred learning style, the learning context, and her relevant domain knowledge. Traditional modeling may not be suitable to capture and adapt to the specific characteristics of the learner, task, and context. This difficulty influences the design of the systems as well as the methods for assessing students’ knowledge and actions.

Tutoring: Metacognitive tutoring is usually done within a context in which students are learning domain-specific skills. This setup requires that the two levels of instruction are integrated in a meaningful way. For example, the design of metacognitive tutors should add metacognitive content without overloading the students’ cognitive capacity, and relevant metacognitive learning goals should be set.

Evaluation: While students’ domain knowledge can be assessed using conventional tests, assessing students’ ability to plan, execute, and monitor their learning is much more challenging. First, this assessment should be independent of students’ domain knowledge. Second, the outcomes of productive metacognitive behavior are often not immediate. They contribute to the quality of the overall learning, but cannot be observed immediately in the solution to a specific problem.

Educational technologies have the potential to tackle these challenges successfully. They offer individual coaching, have the ability to monitor students' progress and learning parameters over extended time periods, and can adapt to individual students' needs. However, it remains largely unknown exactly how educational technologies can help students acquire better metacognitive skills and thereby become better learners with respect to domain-specific skills and knowledge.

This workshop follows earlier workshops on metacognition and SRL (at AIED 2003, AIED 2007, ITS 2008 and ITS2012). In this workshop we discuss the above and other related issues concerning the tutoring of metacognitive and SRL skills using Intelligent Tutoring Systems, focusing on the following: Social self-regulation skills, Scaffolding self-regulation skills and Domain focused self-regulation.

References

1. Aleven, V., & Koedinger, K. R.:, An effective meta-cognitive strategy: learning by doing and explaining with a computer-based Cognitive Tutor. Cognitive Science, 26(2), pp.147-179 (2002)
2. Azevedo, R., Johnson, A., & Chauncey, A. & Graesser, A.:, Use of hypermedia to convey and assess self-regulated learning. In B. Zimmerman & D. Schunk (Eds.), Handbook of self-regulation of learning and performance. New York: Routledge, 102-121 (2011)
3. Bransford, J.: How people learn: brain, mind, experience, and school National Research Council (U.S.). Committee on Learning Research and Educational Practice; National Research Council (U.S.). Committee on Developments in the Science of Learning (2000)
4. Clark, R. C. and Mayer, R.E.: e-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning (2003)
5. Conati C. and VanLehn K.: Toward Computer-Based Support of Meta-Cognitive Skills: a Computational Framework to Coach Self-Explanation . International Journal of Artificial Intelligence in Education, vol 11, pp. 389-415 (2000)
6. Roll, I., Aleven, V., McLaren, B. M., & Koedinger, K. R.:, Improving students' help seeking skills using metacognitive feedback in an intelligent tutoring system. Learning and Instruction, doi:10.1016/j.learninstruc.2010.07.004 (2010)
7. Wagster, J., Tan, J., Wu, Y., Biswas, G., & Schwartz, D.:, Do learning by teaching environments with metacognitive support help students develop better learning behaviors?. In Proceedings of the 29th Annual Meeting of the Cognitive Science Society, pp. 695-700 Nashville, TN. 2007)

.

June, 2013
Amali Weerasinghe, Benedict du Boulay, Gautam Biswas

Program Committee

Co-Chair: Amali Weerasinghe, *University of Canterbury, NZ*
(amali.weerasinghe@canterbury.ac.nz)
Co-Chair: Benedict du Boulay*, University of Sussex, UK*
(b.du-boulay@sussex.ac.uk)
Co-Chair: Gautam Biswas, *Vanderbilt University, USA*
(gautam.biswas@vanderbilt.edu)

Roger Azevedo, *McGill University, Canada*
Ryan Baker, *Worcester Polytechnic Institute, USA*
Paul Brna, *University of Leeds, UK*
Janice D. Gobert, *Worcester Polytechnic Institute, USA*
Neil Heffernan, *Worcester Polytechnic Institute, USA*
Michael J. Jacobson, *University of Sydney, Australia*
Judy Kay, *University of Sydney, Australia*
Susanne Lajoie, *McGill University Canada*
James Lester, *North Carolina State University, USA*
Gordon McCalla, *University of Saskatchewan, Canada*
Amir Shareghi Najar, *University of Canterbury, NZ*
Christina Steiner, *University of Graz, Austria*
Philip Winne, *Simon Fraser University, Canada*
Beverly Woolf, *University of Massachusetts, USA*

Table of Contents

Brief Overview of Social Deliberative Skills[1]

Tom Murray

School of Computer Science
University of Massachusetts Amherst
tmurray@cs.umass.edu

Abstract. Social deliberative skill is the capacity to deal productively with heterogeneous goals, values, or perspectives, especially those that differ from ones own, in deliberative situations. In other papers we describe our team's initial results in exploring this domain, which includes evaluating software features hypothesized to support SD-skills in participants, using machine learning and text analysis methods to recognize SD-skills and other indicators of deliberative quality, and prototyping a Facilitators Dashboard to help third parties get a birds-eye-view of important aspects of an online deliberation so that they can better help participants bring SD-skills to bear within dialogues on controversial topics. In this paper we take the opportunity to expand upon the nature and importance of SD-skills as we currently understand them at a more theoretical level.

Keywords: social metacognition; deliberative dialogue; reflective reasoning; e-learning.

1. Introduction

For about three years our research team has been engaged in studying how to support "social deliberative skills" (SD-skills) in online dialogue (applicable to educational, civic, and workplace contexts). Though the construct of SD-skills overlaps with other skills and capacities, such as metacognition, critical thinking, collaboration skills, and reflective reasoning, it is its own construct, points to an important and understudied area of human capacity, and requires new research to understand it. In other papers we describe our team's initial results in exploring this domain, which includes evaluating software features hypothesized to support SD-skills in participants (Murray et al., 2013a), using machine learning and text analysis methods to recognize SD-skills and other indicators of deliberative quality (Xu et al. 2012, 2103), and prototyping a Facilitators Dashboard to help third parties (facilitators, teachers, mediators, etc.) get a birds-eye-view of important aspects of an online deliberation so that they can better help participants bring SD-skills to bear within dialogues on controversial topics (currently in the context of discussion forums) (Murray et al. 2013b).

[1] Excerpts from a longer paper, in which there are many more references than fit in this extended abstract.

In the discussion section and also in the conference presentation we will summarize our research results, but in this paper we take the opportunity to expand upon the *nature* and *importance* of SD-skills as we currently understand them at a more theoretical level. We also reflect the indeterminacies inherent in defining such psychological constructs.

2. Social Deliberative Skills

The capacity to flexibly and productively negotiate differences of opinion, belief, values, goals, or world-views, is of critical importance in today's world. In the increasingly global world the economic productivity and security of nations can be linked to citizens' and leaders' capacity to understand and deal productively with diverse perspectives. King & Baxter (2005, p. 571) note that "in times of increased global interdependence, producing interculturally competent citizens who can engage in informed, ethical decision-making when confronted with problems that involve a diversity of perspectives is becoming an urgent educational priority…however [these skills] are what corporations find in shortest supply among entry-level candidates."

The capacity to engage skillfully in dialogue with conflicting opinions is important in all realms of social activity including international politics, civic engagement, collaborative work, and mundane familial squabbles. We have coined the term "social deliberative skill" to indicate *the capacity to deal productively with heterogeneous goals, values, or perspectives, especially those that differ from ones own, in deliberative situations.*

Many communication and collaboration interactions now take place on the Internet, which is becoming a ubiquitous global social communication medium. This research investigates how to support the use of social deliberative skills within online communication. Our focus is on supporting mutual understanding and high quality satisfactory outcomes between individuals and/or groups who are communicating with online tools, and much of what we find should be applicable to the support of more skillful deliberation in online work and communication generally. Our overall research goals are to better understand, assess, and support SD-skills in online contexts. We also believe that such skills honed in an online context will partially transfer to other aspects of life. We are interested in investigating online features, tools, and methods that afford, prompt, or gently support SD-skills, rather than teaching them outright.

We differentiate our research from others that focus on argumentation, which aims to help learners generate logical, well-formed, well-supported explanations and justifications. These are certainly important skills, but they are often framed in objective rather than intersubjective (or even ethical) terms. That is, they are about finding the right answer or the most efficient and effective solution to a technical or scientific question—but don't adequately address the specific moments of deliberation or collaboration where opportunities for mutual understanding and mutual recognition arise. They are often studied in the context of problem solving or collaborative work. We also differentiate our work from educational research on creativity, innovation, and collaboration that is framed in terms of pooling ideas and synergizing the best out of them, while often ignoring the skills needed to navigate the challenging straits of controversy, conflict, world-view unfamiliarity, and misunderstanding. We might call the context that we are interested in "difference-motivated social deliberation/inquiry" to highlight the starting point of intersubjective tension. For this research we focus on these social deliberative skills or capacities.

Both the literature on creative problem solving and the literature on civic deliberation emphasize the importance of having diverse perspectives represented in collaborative processes, but scholars on these fields do not always acknowledge the skillfulness needed to work productively with these differences. Meanwhile, in educational research (including educational technology research) there is significant focus on cognitive skills such as metacognition and argumentation, and also considerable research in collaboration, but little work in the specific area addressed by SD-skills.

For this research we will focus on the following social deliberative skills or capacities, which are seen repeatedly in the literature (described using a variety of terms):

1. Social perspective taking (includes cognitive empathy, reciprocal role taking)
2. Social perspective seeking (includes social inquiry, question asking skills);
3. Social perspective monitoring (includes self-reflection, meta-dialogue); and
4. Social perspective weighing (related to "reflective reasoning" and includes comparing and contrasting the available views, including those of participants and external sources and experts).

Capacities implied in the above include: tolerance for uncertainty, ambiguity, disagreement, paradox; and the ability to take first, second, and third-person perspectives on situations or issues (i.e. subjective, intersubjective (you/we/they), and objective).

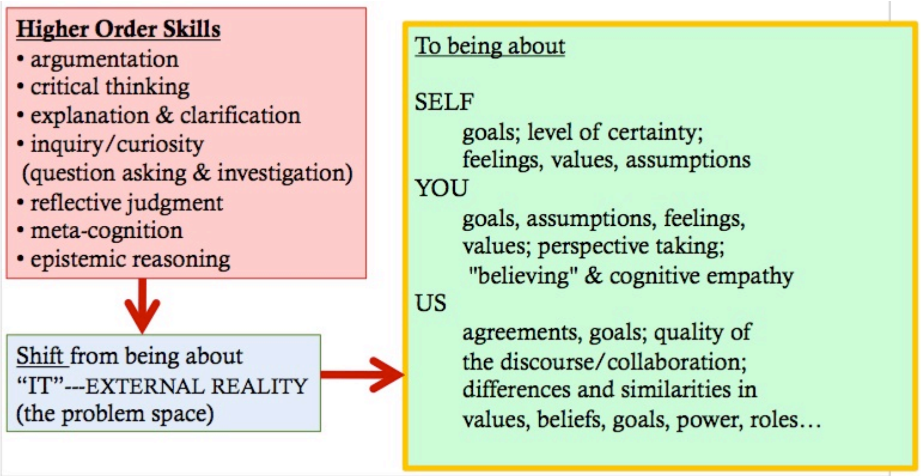
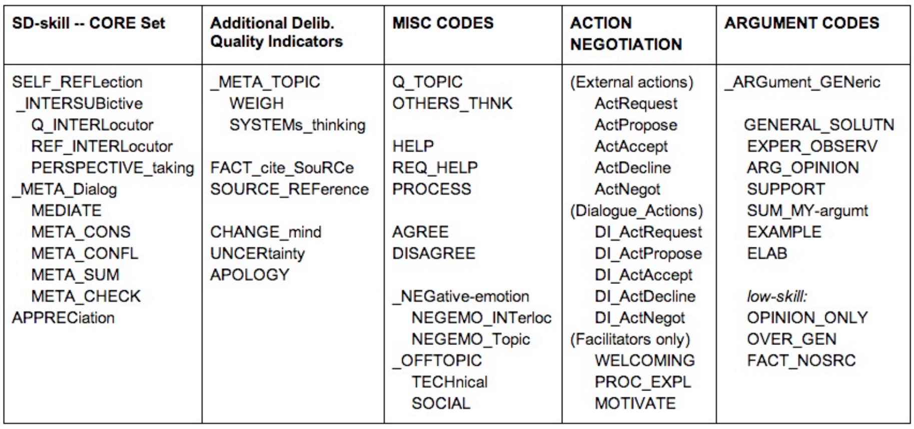

Higher Order Skills
• argumentation
• critical thinking
• explanation & clarification
• inquiry/curiosity
(question asking & investigation)
• reflective judgment
• meta-cognition
• epistemic reasoning
Shift from being about
"IT"---EXTERNAL REALITY
(the problem space)
To being about
SELF
goals; level of certainty;
feelings, values, assumptions
YOU
goals, assumptions, feelings,
values; perspective taking;
"believing" & cognitive empathy
US
agreements, goals; quality of
the discourse/collaboration;
differences and similarities in
values, beliefs, goals, power, roles…

Figure 1: Conceptual Framework for Social Deliberative Skills

Our theoretical frame for these skills is that they involve the *application* of cognitively oriented higher order skills to thinking about the perspectives (or beliefs or arguments) of others (and consequently, of self as well). See Figure 1. When one turns the reflective lens from purely objective ideas about the world toward reflecting on the ideas of specific others (individuals or groups) that one is deliberating with, challenges arise that are beyond the purely cognitive/rational.[2] One is not only reflecting on disembodied ideas but upon *my*/*our*/*your*/*their* ideas. Yet, as forms of reflection, the skills involved are not purely emotional or social. These are critical yet under-explored (and under-supported) moments in collaborative learning, knowledge building, and deliberation in general. Social deliberative skills include reciprocal perspective taking (or cognitive empathy), active perspective seeking (e.g. question-asking skills), self-reflection (e.g. reflecting on one's biases), and meta-dialogue (corrective reflection into the quality of a deliberation or collaboration).

Table 1 illustrates the hand-coding scheme we have been using to code SD-skills.[3] Codes beginning with an underscore are meta-codes subsuming those hierarchically beneath them. Our research on dialogue quality focuses on the first two columns, though we may use codes from other columns as covariates. Though we have defined a number of Argumentation Codes (right column) we do not currently code for them individually (we code them all as ARG_GEN) because, as mentioned, we are interested in intersubjective and reflective skills rather than the argumentation skills per se.



SD-skill -- CORE Set	Additional Delib. Quality Indicators	MISC CODES	ACTION NEGOTIATION	ARGUMENT CODES
SELF_REFLection	_META_TOPIC	Q_TOPIC	(External actions)	_ARGument_GENeric
_INTERSUBictive	WEIGH	OTHERS_THNK	ActRequest	
Q_INTERLocutor	SYSTEMs_thinking		ActPropose	GENERAL_SOLUTN
REF_INTERLocutor		HELP	ActAccept	EXPER_OBSERV
PERSPECTIVE_taking	FACT_cite_SouRCe	REQ_HELP	ActDecline	ARG_OPINION
_META_Dialog	SOURCE_REFerence	PROCESS	ActNegot	SUPPORT
MEDIATE			(Dialogue_Actions)	SUM_MY-argumt
META_CONS	CHANGE_mind	AGREE	DI_ActRequest	EXAMPLE
META_CONFL	UNCERtainty	DISAGREE	DI_ActPropose	ELAB
META_SUM	APOLOGY		DI_ActAccept	
META_CHECK		_NEGative-emotion	DI_ActDecline	*low-skill:*
APPRECiation		NEGEMO_INTerloc	DI_ActNegot	OPINION_ONLY
		NEGEMO_Topic	(Facilitators only)	OVER_GEN
		_OFFTOPIC	WELCOMING	FACT_NOSRC
		TECHnical	PROC_EXPL	
		SOCIAL	MOTIVATE	

Table 1: Text Coding Scheme

This scheme synthesizes prominent frameworks found in the literature (Black et al., 2011; Klein, 2010; Stromer-Galley, 2007; Stolcke et al., 2000) and adds codes for dialogue quality specific to SD-skills. It is most closely related to what has been called "social metacognition" (Salonen et al., 2005; Lin & Sullivan, 2008; Joost et al., 1998; Mischel, 1998). We are in the process of comparing it to King and Kitchener's Reflective Judgment measurement (King & Kitchener, 1994).

[2] Studies of the HOSs in Figure 1 do sometimes include the intersubjective dimension, but the figure highlights how to focus exclusively on it.

[3] Cohen's Kappa Interrater reliability measure for this coding scheme is 71%, (76% agreement) averaged over five dialogue domains we have used it in (this level is considered "good" and is particularly good given the complexity of our coding scheme).

3. Discussion

In this paper (and more in the extended version) we have argued for the importance of studying social deliberative skills, we have differentiated this construct from related ones, and have illustrated how we measure it. We are applying this work to the study of deliberative dialogue in several online domains: classroom discussions of controversial topics, e-commerce and workplace disputer resolution, and civic engagement dialogue. In our studies of how scaffolding features support social deliberative skills we found that reflective tools showed a significant difference with large effect size (Murray et al. 2013a). We have made progress in using text analysis tools (CohMetrix, Graesser et al. 2010) and LIWC (Pennabaker et al. 2007) and machine learning algorithms to categorize social deliberative skill automatically (see Xu et al. 2012, 2013).

References

Black, L., Welser, H., Cosley, D., and DeGroot, J., Self (2011). Governance Through Group Discussion in Wikipedia Measuring Deliberation in Online Groups. Small Group Research 42(5) pp. 595-634.

Graesser, A., & McNamara, D. (2010). Computational analyses of multilevel discourse comprehension. Topics in Cognitive Science 3(2), 371–398. 2010.

Jost, J. T., Kruglanski, A. W., & Nelson, T. O. (1998). Social metacognition: An expansionist review. Personality and Social Psychology Review, 2(2), 137-154.

King, P. M. & Baxter Magolda, M. (2005). A developmental model of intercultural maturity. Journal of College Student Development, 46 (6), 571-592.

King, P.M. & Kitchener, K.S. (1994). Developing reflective judgment: Understanding and promoting intellectual growth and critical thinking in adolescents and adults. Jossey-Bass.

Klein, M. (2010). Using Metrics to Enable Large-Scale Deliberation. Collective Intelligence In Organizations: A Workshop of the ACM Group 2010 Conference. Sanibel Island, Florida, USA.

Lin, X. & Sullivan, F. (2008). Computer contexts for supporting metacognitive learning. In J. Voogt, G. Knezek (eds.) International Handbook of Information Technology in Primary and Secondary Education, 281–298. Springer Science+Business Media, LLC.

Murray, T., Stephens, A.L., Woolf, B.P., Wing, L., Xu, X., & Shrikant, N. (2013a). Supporting Social Deliberative Skills Online: the Effects of Reflective Scaffolding Tools. Proceedings of **HCI** International 2013, July, 2013, Las Vegas.

Murray, T., Wing, L., Woolf, B., Wise, A., Wu, S., Clark, L. & Osterweil, L. (2013b). A Prototype Facilitators Dashboard: Assessing and visualizing dialogue quality in online deliberations for education and work. Submitted to 2013 International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government.

Pennebaker, J. W., Chung, C. K., Ireland, M., Gonzales, A. L., & Booth, R. J. (2007). The development and psychometric properties of LIWC2007. Austin, TX: www.LIWC.net.

Salonen, P., Vauras, M., & Efklides, A. (2005). Social Interaction--What Can It Tell Us about Metacognition and Coregulation in Learning?. European Psychologist, 10(3), 199.

Stolcke, A., Ries, K., Coccaro, N., Shriberg, J., Bates, R., Jurafsku, D., et al. (2000). Dialogue act modeling for automatic tagging and recognition of conversational speech. Computational Linguistics, 26(3), 39– 373.

Stromer-Galley, J. (2007). Measuring Deliberation's Content: A Coding Scheme. Journal of Public Deliberation, 3(1).

Xu, X., Murray, T., Smith, D. & Woolf, B.P. (2013) . If You Were Me and I Were You Mining Social Deliberation in Online Communication. Proceedings of EDM-13, Educational Data Mining, July, 2013, Memphis, TN.

Enhancing socially shared regulation in working groups using a CSCL regulation tools

Ernesto Panadero [1], Sanna Järvelä [1], Jonna Malmberg [1], Marika Koivuniemi [1],
Chris Phielix [2], Jos Jaspers [2] & Paul Kirschner [3]

1 Faculty of Education, University of Oulu, Finland
{ernesto.panadero , sanna.jarvela , jonna.malmberg , marika.koivuniemi}@oulu.fi
2 Educational Sciences, University of Utrecht, The Netherlands
{C.Phielix , J.G.M.Jaspers}@uu.nl
3 Centre for Learning Sciences & Technologies CELSTEC, Open University, The Netherlands
Paul.Kirschner@ou.nl

Abstract. Socially shared regulation of learning (SSRL) refers to processes by which group members collectively regulate activity within a balanced shared responsibility model. SSRL has shown to increase performance and learning when compared to other forms of regulating collaborative work (co-regulation). SSRL, however, is a relatively new concept which needs empirical study, especially in how to promote this it in real learning settings. This study is a major first step, studying the promotion of SSRL through an often used online collaborative work environment augmented with three SSRL tools (Radar, OurPlanner, OurEvaluator) to stimulate and enhance the four self-regulatory phases of learning: planning, monitoring, evaluating and regulating. Through the use environment and tools, students will be better able to share regulation of collaborative learning.

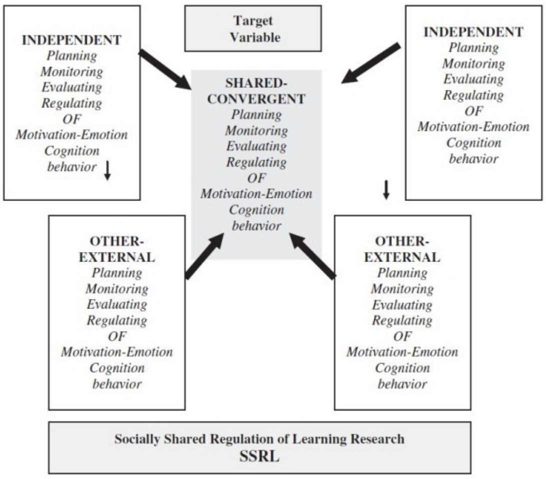
Keywords: Self-regulated learning, socially shared regulation, collaborative work, CSCL, regulation tools, scaffolding.

1 Theoretical framework

Regulation of learning has traditionally explored individual characteristics in various learning situations (self-regulation; [1]). However, new learning demands involving collaborative learning situations has shifted the focus towards the social aspects of regulated learning, namely co-regulation and socially shared regulation of learning [4] [6]. Co-regulation of learning refers to processes where a group collaborates under unbalanced regulation (e.g. one of the members exerting power and deciding what to do). Socially shared regulation of learning (SSRL) refers to processes where group members collectively regulate activity; where decisions and regulatory activities are decided in shared ways. Research has shown that SSRL can produce better learning outcomes and enhance performance [5] [8]. Collaborative learning interventions, thus, should aim at promoting SSRL.

As can be seen in Figure 1, SSRL is reached through a number of iterations between the group members' individual self-regulation and the others self-regulation, until shared-convergent regulation is achieved [4]. As with individual self-regulation, the group's shared regulation is composed of four recursive phases: planning, monitoring, evaluating and regulating [9]. During the planning phase, the group establishes its goals and standards, and organizes the actions they will need to make to complete the task. While monitoring, group members compare the procedure they are following with the initial plan of action and the goals for the activity. Evaluating implies that the students compare the fit of their product to the standards determined in the planning phase. Finally, group members enter the regulating phase in which they make the changes needed to overcome an eventual gap between the standards set and the final product achieved.

Figure 1. Socially Shared Regulation of learning (extracted from [4]).



Research in the individual self-regulation field has found that interventions should aim to promote planning, monitoring and evaluating and that the most successful interventions are composed of an array of aspects: cognitive, motivational and emotional [3]. Research on promoting SSRL is limited necessitating building on research on individual learning [2]. The key aspect is that, to promote SSRL in the groups, a shared space is needed in which members can collaborate, creating and deciding how to regulate their efforts and actions. In a practical sense, this implies creating tools that target the phases of regulated learning such that students are able and stimulated to plan together, monitor how the group is performing, evaluate the final product against the standards set up at the beginning and, finally regulate/change accordingly to achieve their learning goals [6]. This is to say, prompt the aspects of socially shared regulation which often are salient for the students.

With these key aspects in mind, we tailored an operating online environment in which we could promote socially shared regulation. The Virtual Collaborative Research Institute (VCRI) (http://edugate.fss.uu.nl/~crocicl/vcri_eng.html) is an online tool to promote collaborative work, usually with group members work on their own computer, either synchronously or asynchronously [7]. In the PROSPECTS project (https://let.drupal.oulu.fi/en/node/10135), the VCRI environment was used as a platform to set up and promote SSRL by plugging in existing features of that environment such as Radar, Co-Writer and chat.

Radar is a tool with which group members report about aspects of their individual self-regulation relevant for the collaborative work (e.g., I know how to perform the task), and aspects related to the group work (e.g., I think the group is capable of performing the task). Students rate these aspects along six different axes in a five Likert scale yielding a radar-diagram. The six items in the axes are: (1) I understand the task, (2) I know how to do this task, (3) This task is interesting, (4) My feelings influence on my working, (5) I feel capable of doing this task, and (6) My is capable of doing this task. The idea behind Radar is that students will be aware of their strengths and weaknesses in a current situation and thus the group will be aware of their strengths and weaknesses that they might confront during the task assignment.

Co-writer, a shared writing space, was divided to promote collaborative planning (OurPlanner), serve as a platform for the students on-line task execution

(Task execution) and finally, promote collaborative evaluation of the regulated learning (OurEvaluator). OurPlanner is a shared new tool which prompts the students in their planning (e.g., describing the task, describing its purpose, creating a concrete plan). Task execution is the place where group members can collaboratively write and modify their course assignments. Finally, OurEvaluator allows group members together evaluate and regulate aspects of their collaboration. The idea behind these tools is to help students collaboratively clarify the goals and standards for the task, along with the procedure and strategies they will use. What they write in the Co-writer should be used to guide their monitoring and evaluating.

2 Procedure

First year teacher education students (N = 130) are participating in a 'Multimedia as a learning project' course. The course consists of nine sessions where the students worked collaboratively in 3-4 member groups. Each learning session is divided in two different parts: (1) a face to face part at the university computer class with teacher support, and then (2) an online part that students perform individually. In both phases the SSRL tools is actively used.

The face to face sessions have three phases. First, the instructor introduces the task. Then, the students individually complete the Radar and as a result see each other's Radars. This is followed by the groups collaboratively planning their work on the assignments (goals, strategies, etc.) using OurPlanner. The conversations during this planning are recorded. In the third phase, they work together performing the task.

The online sessions share the similar procedure as face to face sessions with one extra phase and with the students use the full SSRL regulation tool resources of the VCRI environment working synchronously on their own computer at home or at the university. First, the assignment is presented in VCRI. Then, teams plan their goals and the organization of the assignment using OurPlanner and negotiating through the chat. Third, they perform the task online using chat for negotiation during the task execution. Finally, they evaluate their work using the OurEvaluator.

In sum, the intervention promotes SSRL through the different phases. The planning of collaborative work is conducted during the planning phases in both face to face and online. Students monitor their progress during the working phases. Evaluating and regulating happens when students receive the online task instructions –being able to reflect about what they have achieved so far- and, of course, during the evaluation phase of the online session once the task is done. What VCRI adds is the collaboration tool: allowing the students to work together and regulate through its uses.

3 Results

The first notions of the data show promising findings dealing with the SSRL tool's prompting not only socially shared regulation, but also collaborative learning. The VCRI environment data will be analyzed looking for traces of SSRL to classify groups according to their regulation and performance. The data collection is currently ongoing, but the preliminary findings will be presented at the workshop.

4 References

1. Boekaerts, M., & Corno, L. (2005). Self-regulation in the classroom: A perspective on assessment and intervention. *Applied Psychology-an International Review-psychologie Appliquee-revue Internationale, 54*(2), 199-231.
2. Dignath, C., & Büttner, G. (2008). Components of fostering self-regulated learning among students. A meta-analysis on intervention studies at primary and secondary school level. *Metacognition and Learning, 3*, 231-264. doi: 10.1007/s11409-008-9029-x
3. Dignath, C., Büttner, G., & Langfeldt, H. (2008). How can primary school students learn self-regulated learning strategies most effectively? A meta-analysis on self-regulation training programmes. *Educational Research Review, 3*(2), 101-129. doi: 10.1016/j.edurev.2008.02.003
4. Hadwin, A. F., Järvelä, S., & Miller, M. (2011). Self-regulated, co-regulated, and socially shared regulation of learning. In B. J. Zimmerman & D. H. Schunk (Eds.), *Handbook of self-regulation of learning and performance* (pp. 65-84). New York: Routledge.
5. Janssen, J., Erkens, G., Kirschner, P. A., & Kanselaar, G. (2012). Task-related and social regulation during online collaborative learning. *Metacognition and Learning, 7*(1), 25-43. doi: 10.1007/s11409-010-9061-5
6. Järvelä, S., & Hadwin, A. F. (2013). New frontiers: Regulating learning in CSCL. *Educational Psychologist, 48*(1).
7. Phielix, C. (2012). *Enhancing collaboration through assessment & reflection.*
8. Volet, S., Summers, M., & Thurman, J. (2009). High-level co-regulation in collaborative learning: How does it emerge and how is it sustained? *Learning and Instruction, 19*(2), 128-143. doi: 10.1016/j.learninstruc.2008.03.001
9. Winne, P. H., & Hadwin, A. F. (1998). Studying as self-regulated engagement in learning. In D. Hacker, J. Dunlosky & A. Graesser (Eds.), *Metacognition in educational theory and practice* (pp. 277-304). Hillsdale, NJ: Erlbaum.

How should SE be supported – during problem solving or separately?

Amali Weerasinghe, Amir Shareghi Najar, Antonija Mitrovic

Intelligent Computer Tutoring Group (ICTG)
University of Canterbury, New Zealand
(amali.weerasinghe, tanja.mitrovic,
amir.shareghinajar)@canterbury.ac.nz

Abstract. Self-explanation (SE) has proven to be an effective meta-cognitive strategy. However, some performance-oriented students tend to not take advantage of the SE opportunities provided as they are seen as extra work that does not directly contribute to problem solving. We focus on approaches that can be used to motivate such students to take advantage of SE support. As a first step, we analysed SE support provided in some systems and discuss their limitations. We also outline a study that compares the two approaches: separating SE support from problem solving versus interleaving the two.

1 Introduction

Self-explanation (SE) has proven to be an effective meta-cognitive strategy. Bransford et al. [1] suggest focusing on metacognition as one of three principles that should be applied to educational research and design, as stated in the influential volume "How People Learn". According to previous research studies, only a few students self-explain spontaneously, and therefore SE prompts have been used to encourage students to explain instructional material to themselves [2]. SE prompts can be of different types, according to the knowledge they focus on. For instance, Hausmann et al. [3] compared *justification-based prompts* (e.g. "what principle is being applied in this step?") and *meta-cognitive prompts* (e.g. "what new information does each step provide for you?") with a new type called *step-focused prompts* (e.g. what does this step mean to you?"). They found that students in the step-focused and justification conditions learnt more from studying examples than students in the meta-cognitive prompts condition. In another study, Chi and VanLehn [4] categorised SE as either procedural explanation (e.g. answer to "Why was this step done"), or derivation SE (e.g. answer to "where did this step come from?"). In [5], SE prompts are categorized into *procedural-focused self-explanation* (P-SE) prompts and *conceptual-focused self-explanation* (C-SE) prompts. P-SE prompts were given after examples to assist students to focus on procedural knowledge as the examples have shown to increase conceptual knowledge. On the other hand, after solving problems, students were given C-SE prompts in order to help the students to gain the corresponding conceptual knowledge covered in the problems they just completed.

SE has generally been supported in the context of a problem-solving environment. Even though many systems use the problem-solving context, they include additional steps to support SE. For instance, an enhanced version of Geometry Explanation Tutor expects students to explain every problem-solving step [6]. Asking students to explain each step is an additional task in the typical problem-solving process. How a student interacts with the learning environment depend on his/her attitude and learning goals [7]. If a student has a performance-oriented focus (i.e. attempting to demonstrate their ability by completing as many problems as they can without paying much attention to acquiring knowledge), it is possible that they may view this as extra work. In such situations, do we keep including such opportunities anyway to support SE as it is beneficial for students' leaning? This decision may have a negative impact as the student may be demotivated and likely to be disengaged from the learning. The other alternative is to provide only problem-solving support and support SE when they become more proficient; are students less likely to take advantage of SE opportunities when they are novices?

As a first step towards exploring these questions, we analysed the SE support provided by different systems. The way these systems support SE can be categorized as separating SE from problem solving vs interleaving the two. The systems in the first category provide SE opportunities immediately after a problem/step is completed. This may also result in disengagement from taking advantage of a learning opportunity as they have completed the problem/step and want to move to the next problem/step. Interleaving SE support with problem solving expect students to self-explain during problem solving. Will the students be more motivated if these opportunities to self-explain are integrated with problem-solving? What is the effect of each approach on student's mental model of process of problem-solving i.e. if the integrated approach is used, will the students feel that SE is a vital ingredient of learning by solving problems and vice versa. Exploring these issues will provide us with initial insights about students' behaviour towards SE support. This will enable us to design ITSs that dynamically adapt their pedagogical decisions such as SE support not only on the individual student's competency of the instructional task, but also on their learning goals.

In this paper we discuss some studies that use one of the two strategies (integrated approach vs. separation approach) and our plans to conduct an evaluation study that compares these two approaches.

2 Interleaving SE support with problem solving

We now discuss two systems that interleave SE support with problem solving. Both these systems expect students to provide self-explain during problem-solving.

2.1 Geometry Explanation Tutor

A new version of the Geometry Explanation Tutor was created to provide support for SE while students learn about the properties of angles in various kinds of diagrams [6]. In addition to solving problems, students were expected to explain all the steps for each problem. For example, a student could explain a step in which the triangle sum theorem was applied by typing "Triangle Sum". A Glossary of geometry knowledge was provided as a way of helping students to provide self-explanations. The Glossary lists relevant theorems and definitions, illustrated with short examples. It is meant to be a reference source which students can use freely to help them solve problems. Students could enter explanations by selecting a reference from the Glossary or could type their explanations. The tutor provided feedback on the students' solutions as well as their explanations. Further, it provided on-demand hints, with multiple levels of hints for each step. SE is supported via the additional task of explaining each problem-solving step: the students were expected to solve each step in a problem and provide explanations at the same time. Hence this system supports SE during problem solving, but support is provided using an additional task. As the SE is not adaptive, students may have to specify a theorem multiple times for a problem, if it has been used in several steps within the problem.

A study was conducted to compare the performances of students when they explain their problem-solving steps in their own words with their peers who did not. The students who explained the problem-solving steps learnt with greater understanding compared to their peers who did not. The explainers were also more successful on transfer problems.

2.2 NORMIT-SE

NORMIT, an ITS that teaches data normalization, was enhanced to support SE [8]. The enhanced system, NORMIT-SE, expects an explanation for each action type performed for the first time. For the subsequent actions of the same type, explanation is required only if the action is performed incorrectly. This approach would reduce the burden on more able students (by not asking them to provide the same explanation every time an action is performed correctly), and also that the system would provide enough situations for students to develop and improve their explanation skills.

Students provide explanations by selecting one of the offered options. The order in which the options are given is random, to minimize guessing. For example, if the specified candidate key is incorrect, NORTMIT-SE asks the following question "This set of attributes is a candidate key because……:"

If the student's explanation is incorrect, he/she will be given another question, asking to define the underlying domain concept (i.e. candidate keys). An example of such a question is "A candidate key is………… ". In contrast to the first question, which was problem-specific, the second question focuses on domain concepts. If the student selects the correct option for a question, he/she can resume problem solving. If the student's answer is incorrect, NORMIT will provide the correct definition of the concept.

An evaluation study was conducted to investigate the effect of explaining problem-solving steps on both procedural and conceptual knowledge [8]. The students in the experimental group were expected to explain their problem-solving steps while their peers in the control group just solved problems. The experimental group acquired knowledge (represented as constraints) significantly faster than the control group. There was no significant difference between the two conditions on the post-test performance, and it might be due to the short duration of their sessions interacting with the system. Furthermore, the analysis of the self-explanation behavior shows that students find problem-specific question (i.e. explaining their action in the context of the current problem state) more difficult than defining the underlying domain concepts.

3 Separating SE support from problem solving

SQL-Tutor is an ITS that teaches database querying and was enhanced to provide SE support after each problem was completed [5]. The students were expected to solve the given problems as in the original version of SQL-Tutor which provided multiple levels of feedback. Upon completion of a problem, students were given an opportunity to self-explain. The student received a C-SE prompt with multiple options from which the correct one has to be selected. "What does DISTINCT in general do??" is an example of a C-SE prompt. There was only one SE prompt per problem. The prompts were non-adaptive and depended only on the problem. As the SE support focused only on conceptual knowledge, the problem-solving context does not have to be used to support SE.

A study was conducted to investigate the effects of such SE support on student learning. This was a part of a larger study and we report only the relevant results. Problems were provided in pairs. i.e. students solved two isomorphic problems in each pair. The participants were 12 students enrolled in an introductory database course at the University of Canterbury. Participants were informed that they would see ten pairs of problems, and that the tasks in each pair were similar. Providing this information to students may have motivated them to use problem pairs more efficiently. Analysis revealed that students performance on the post-test was significantly higher in comparison to the pre-test performance ($p<.01$).

4 Discussion and Future Work

The three research attempts discussed can be categorized using different criteria such as the type of approach used, the type of SE supported and the target instructional task. Both the enhanced Geometry Explanation Tutor and NORMIT-SE provide SE support during problem-solving. In contrast, SQL-Tutor provides SE support after problem solving. Furthermore, NORMIT-SE provides both conceptual and procedural SE. In contrast, the other two systems use only conceptual prompts.

The only system that provides adaptive SE support is NORMIT-SE. However, NORMIT-SE does not consider the learning goals of each student to customise SE support. However we believe that SE support could be more effective when it is customized based on both a learner's knowledge and learning goals. Such customising has the potential to motivate students to take advantage of SE support instead of burdening them.

In order to explore how students utilise the different ways of SE support, we plan to conduct a study within the context of NORMIT-SE with four groups. All the groups will be asked to solve several problems while receiving typical feedback with multiple levels of help from NORMIT-SE. Groups 1 and 2 will be given conceptual SE-prompts and the other two (groups 3 and 4), procedural prompts. Groups 1 and 3 will be asked to self-explain after a problem is completed. The remaining two groups (groups 2 and 4) will self-explain when they submit their first attempt for a problem. We hypothesise that providing conceptual prompts at the end of each problem or procedural prompts after the first attempt are more beneficial than the other two scenarios. We also plan to identify measures related to a student's problem-solving behavior to infer learning goals for each student. Such measures can include the number of times a student access the full solution, number of times each help level is accessed and the number of times help is sought for a problem. Based on this analysis, we plan to classify students as having a performance-oriented or a learning-oriented focus. This classification will enable us to design ITSs that dynamically adapt SE support not only on the individual student's competency of the instructional task, but also on their learning goals

References

1. Bransford, J. (2000). How people learn: brain, mind, experience, and school National Research Council (U.S.). Committee on Learning Research and Educational Practice; National Research Council (U.S.). Committee on Developments in the Science of Learning.
2. Chi, M.T.H., De Leeuw, N., Chiu, M.H., LaVancher, C. (1994). Eliciting self-explanations improves understanding. Cognitive science. 18, 439–477.
3. Hausmann, R., Nokes, T., VanLehn, K., Gershman, S. (2009). The design of self-explanation prompts: The fit hypothesis. Proc. 31st Annual Conference of the Cognitive Science Society. pp. 2626–2631.
4. Chi, M.T.H., VanLehn, K.A. (1991). The content of physics self-explanations. The Journal of the Learning Sciences. 1(1), 69–105.
5. Shareghi Najar, A., Mitrovic, A. (2013). Examples and Tutored Problems: How can Self-Explanation Make a Difference to Learning, 16th International Conference on AI in Education.
6. Aleven, V., Koedinger, K.R. (2002). An effective metacognitive strategy: Learning by doing and explaining with a computer-based Cognitive Tutor. Cognitive Science. 26, 147–179.
7. Arroyo, I., Woolf, B. (2005). Inferring learning and attitudes from a Bayesian Network of log file data. In: Proceedings of the 12th International Conference on Artificial Intelligence in Education, 33–40.
8. Mitrovic, A. (2005). The Effect of Explaining on Learning: a Case Study with a Data Normalization Tutor, 12th International Conference on Artificial Intelligence in Education.

An Investigation of Successful Self-Regulated-Learning in a Technology-Enhanced Learning Environment

Christina M. Steiner[1], Gudrun Wesiak[1,2], Adam Moore[3], Owen Conlan[3]
Declan Dagger[4], Gary Donohoe[5], & Dietrich Albert[1,2]

[1] Knowledge Technologies Institute, Graz University of Technology, Austria
{christina.steiner,gudrun.wesiak,dietrich.albert}@tugraz.at
[2] Department of Psychology, University of Graz, Austria
{gudrun.wesiak,dietrich.albert}@uni-graz.at
[3] KDEG, School of Computer Science and Statistics, Trinity College, Dublin, Ireland
{mooread,owen.conlan}@scss.tcd.ie
[4] EmpowerTheUser, Trinity Technology & Enterprise Campus, The Tower, Dublin, Ireland
declan.dagger@empowertheuser.com
[5] Department of Psychiatry, School of Medicine, Trinity College, Dublin, Ireland
donoghug@tcd.ie

Abstract. Self-regulated learning (SRL) and metacognition are key in the context of 21st century education, adult training, and lifelong learning. For instructional strategies to foster metacognition and self-regulation it is crucial to know what are good metacognitive and SRL behaviors. We investigated this question in the context of a training simulator in a curriculum setting with 152 medical students. Learning behavior and personal attributes were examined in relation to metacognitive awareness. The results on characteristics of successful SRL confirm findings from traditional learning settings for a TEL context.

Keywords: self-regulation, metacognition, expert learner, training simulator.

1 Introduction

Broad interest in metacognition and self-regulated learning (SRL) can be identified in current research, as well as educational practice [1]. Often used synonymously, they are considered as mutual core components of learning. Learners highly skilled in those aspects are often referred to as 'expert learners' [2][3]. Given the demands of 21st century education, adult training, and lifelong learning; taking responsibility for one's own planning, performing, monitoring, and regulating learning is crucial. In particular, for technology-enhanced learning (TEL), SRL and metacognition are recognized as having a key role [4]. It is acknowledged that SRL and metacognitive processes require the availability of appropriate knowledge and strategies. Learners need support in acquiring and applying these skills; accordingly, this area and related intervention programs are intensely investigated [5]. For sound instructional and scaffolding strategies an in-depth understanding of *good* metacognitive and SRL behaviors is crucial [3]. This paper investigates characteristics of successful SRL in the scope of learning episodes with an immersive experiential training simulator.

2 What is Good SRL Behavior?

Successful (and less successful) learning is not about the question of whether self-regulation and metacognition occur – all learners think about and try to regulate their learning in some way, but there are dramatic differences in how they approach it. A high quality and quantity of self-regulatory and metacognitive processes goes along with better learning performance and achievements [6][7]. Research has attempted to identify the differences between lower and higher achieving learners to draw implications for SRL and metacognitive scaffolding and strategy training [3][8]. Expert learners know, and successfully employ, more and better cognitive and metacognitive strategies [2][6]. A variety of personal attributes were found to characterize and distinguish students with high versus low metacognitive and SRL abilities (see *e.g.* [1][8] for an overview). Effective learning is related to higher levels of motivation and self-motivational beliefs [6]; whereas underachievers are known to be less efficacious about their learning and to have a lower self-esteem, to be more impulsive, and to give up earlier and more easily. In particular, they are also more anxious and fear failure [8]. The research aiming at explaining why some learners are more successful than others so far has been concentrated on traditional learning situations. TEL environments, such as web-based courses, impose additional demands on learners [9]. It is therefore important to examine the characteristics of effective metacognition and SRL more directly in a TEL context, to see whether the results confirm the state of the art from traditional learning settings and to identify whether there are any peculiarities for TEL. This paper presents an empirical investigation pursuing that goal. One main objective was to investigate SRL behavior and learner characteristics in relation to learners' general metacognitive awareness.

3 An Empirical Study in an Experiential Learning Environment

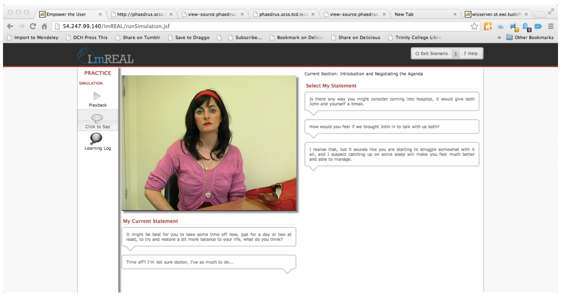
3.1 Method

Augmented Training Simulator. ETU's[1] RolePlay Simulation Platform offers simulation scenarios teaching student doctors about effective doctor-patient communication (see Figure 1). Users' main task is to select appropriate dialogues for clinical interviews with patients diagnosed with either mania or depression. The TEL environment embeds a range of features to support self-regulation. More specifically, the simulator provides learning triggers for delivering targeted in-context coaching, behavioral feedback and strategic reflections to reinforce learning and aid transfer to the job. The platform also doubles as a psychometric profiling, behavioral measurement and skill assessment tool. Metacognitive scaffolding was provided to learners within the ETU simulator using calls to a RESTful service developed as part of the ImREAL project[2]. The service utilizes a cognitive model to support self-reflection and presents items from the Metacognitive Awareness Inventory (MAI) [7],

[1] www.etu.ie

[2] www.imreal-project.eu

e.g. “Have you focused your attention on the important information?”. It has previously been shown that providing this scaffolding within the ETU platform is beneficial [10]. Alongside the scaffolding thinking prompt is an open text box for collecting reflection notes which is consistently prefaced with a short text: “Reflect now on your learning: Was this last part of the simulation useful for you?” In addition, there is a place to reflect in the simulator’s note-taking tool, where learners can record and share notes.



Current Section: Introduction and Negotiating the Agenda

Select My Statement

Is there any way you might consider coming into hospital, it would give both John and yourself a break.

How would you feel if we brought John in to talk with us both?

I realise that, but it sounds like you are starting to struggle somewhat with it all, and I suspect catching up on some sleep will make you feel much better and able to manage.

My Current Statement

It might be best for you to take some time off now, just for a day or two at least, to try and restore a bit more balance to your life, what do you think?

Time off? I'm not sure doctor, I've so much to do...

Figure 1. Screenshot of the ETU RolePlay Simulation Platform.

Participants, Instruments, and Procedure. In spring 2013, 152 third year medical students (*M* = 22.81 years old, *SD* = 3.79) from Trinity College Dublin participated in the study as part of their medical curriculum. A mixed-method approach capturing metacognition and SRL in terms of users’ general learning approach (self-report) and the actual activities during simulator usage (log data) was applied [11]. Students completed a cohort characterization survey before interacting with the simulator. Besides demographic questions and a personality questionnaire (SSP, Swedish Universities Scales of Personality [12]), a standard scale assessing metacognitive awareness (MAI [7]) was administered. Students could then use the simulator as long and often they wished. Interaction data and text entries from reflection notes and the note-taking tool were tracked by the simulator and served for investigating learning behavior. Self-predicted and objective learning performances based on an assessment of interview skills built into the simulator were also used. This trace methodology corresponded to the idea of examining SRL as a process [13]. After the learning episode students provided feedback on learning with the simulator in a survey covering the perception of reflection prompts, motivation, and SRL (QSRL, [14]).

3.2 Results

Log data from 152 students performing the training in the simulator was available, whereas subsamples of 76 (MAI) and 85 (SSP) filled out the *pre*-questionnaire and only 39 (prompts), 25 (QSRL) and 29 (motivation) students completed the *post*-survey. Samples sizes for filling out both the MAI (as grouping variable) and one of the other questionnaires (as dependent variable) were even smaller. To investigate differences with respect to learning activities and feedback on the simulator between users with high and lower metacognitive awareness (and thus SRL-abilities), the subsample that had completed the MAI *before* entering the simulator was split at the median into two groups. Focusing on SRL as a process [13], this was done using the regulation of cognition (ROC) subscales and scores (*$Md_{MAI\text{-}ROC}$* =.69; *$M_{low\text{-}ROC}$* =.56, *SD*=.13; *$M_{high\text{-}ROC}$* =.83, *SD*=.08), which address the metacognitive strategies and subprocesses of learning [7].

Independent samples t-tests for high (high ROC) and low (low ROC) metacognitive awareness revealed significant differences (all p<.05) regarding participants' SRL-behavior, personality traits, motivation, as well the number of notes taken during the interview training (see Figure 2). More specifically, students with higher metacognitive awareness (as far as the regulation of knowledge is concerned) are also better in monitoring their own learning processes ($t_{(18)}$ = -2.15), have higher achievement motivation ($t_{(18)}$ = -2.26), attribute their successes more strongly to their abilities ($t_{(18)}$ = -2.88), and are more motivated regarding their current learning situation ($t_{(26)}$ = -2.83), especially to apply what they have just learned. Additionally they took more notes during the interview training (with N=14 and no equal variances: $t_{(9)}$ = -2.38), i.e. they reflected more explicitly on the decisions they made during the training. On the other hand, they show lower trait anxiety ($t_{(70)}$ = 2.04) and lower scores on lack of assertiveness ($t_{(70)}$ = 2.7). There was no difference regarding the perception of thinking prompts. Both groups rated them as helpful and appropriate on 5-pt scales (for 10 questions all *Md* = 4, overall *M* = 3.6, *SD* = .58).

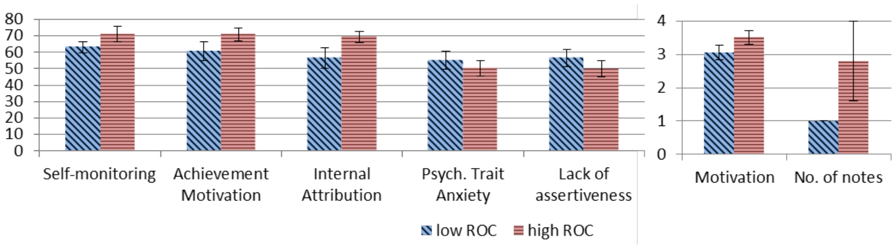


Figure 2. Mean SRL scores, personality traits, motivation, and number of notes for low and high metacognitive awareness.

4 Conclusion

The outcomes of the presented study argue for the transferability of known characteristics of good metacognition and SRL identified in traditional learning settings to a TEL context. Although comparisons are actually based on groups of high vs. medium metacognitive abilities, a range of distinguishing differences could be identified. In line with previous results that expert learners apply more metacognitive strategies, high ROC students were shown to more extensively monitor and evaluate their own learning and to take more notes in the simulator. Also a trend of higher learning performance (ETU score) being associated with higher SRL abilities was found: Results revealed higher SRL scores on all nine QSRL subscales for better performing students in the simulation (N = 25). However, since these differences are not statistically significant, further research with larger samples is necessary.

No difference was found in students' abilities of predicting their own performance. A general novelty effect of the learning setting might have mitigated an expected difference in persistence in terms of duration of simulator usage. Since achievement motivation refers to the desire to perform well on challenging tasks and is evidenced by effort and persistence, though, the higher scores identified for the high ROC group may be related to previous results on higher persistence of expert learners. This group also reported a higher motivation to transfer the just acquired skills to real world interviews. The lower internal attribution of success found for low ROC resembles existing results on lower self-efficacy for learners with low metacognitive abilities. In addition, low ROC students were shown to be more anxious, confirming previous results on higher anxiety for lower skilled learners. Follow-up investigations with samples featuring a higher range in metacognitive and SRL abilities are planned.

Acknowledgments. The research leading to these results has received funding from the EU 7th Framework Programme under grant ICT 257831 (ImREAL project).

References

1. Duckworth, K., Akerman, R., McGregor, A., Salter, E., Vorhaus, J.: Self regulation: A review of literature. Report 33. Centre for Research on the Wider Benefits of Learning, Institute of Education, London (2009)
2. Ertmer, P.A., Newby, T.Y.: The expert learner: Strategic, self-regulated, and reflective. Instructional Science 24 (1996) 1--24
3. Ley, K., Young, D.B.: Instructional principles for self-regulation. Educational Technology Research and Development 49 (2001) 93--103
4. Dettori, G., Persico, D.: Fostering self-regulated learning through ICT. IGI Globak, Hershey (2011)
5. Steffens, K.: Self-regulated learning in technology-enhanced learning environments: Lessons of a European peer review. European Journal of Education 41 (2006) 353--379
6. Zimmerman, B.: Becoming a self-regulated learner: An overview. Theory into Practice 41 (2002) 64--70
7. Schraw, G., Dennison, S.R.: Assessing metacognitive awareness. Contemporary Educational Psychology 19 (1994) 460--475
8. Cubukcu, F.: Learner autonomy, self-regulation and metacognition. International Electronic Journal of Elementary Education 2 (2009) 53--64
9. Narciss, S., Proske, A., Koerndle, H.: Promoting self-regulated learning in web-based learning environments. Computers in Human Behavior 23 (2007) 1126--1144
10. Berthold, M., Moore, A., Steiner, C., Gaffney, C., Dagger, D., Albert, D. et al.: An initial evaluation of metacognitive scaffolding for experiential training simulators. In Ravenscroft, A., Lindstaedt, S., Delgado Kloos, C., Hernández-Leo, D. (eds.) 21st century learning for 21st century skills. LNCS, vol. 7563, pp. 23--36. Springer, Berlin (2012)
11. Steiner, C.M., Berthold, M., Albert, D.: Evaluating the benefit of a learning technology on self-regulated learning. A mixed method approach. Fourth Workshop on Self-regulated learning in Educational Technologies (SRL@ET). ITS Conference, Crete, Greece (2012)
12. Gustavsson , J.P., Bergman H., Edman, G., Ekselius, L., von Knorring, L., Linder, J.: Swedish universities Scales of Personality (SSP): construction, internal consistency and normative data. Acta Psychiatrica Scandinavica 102 (2000) 217--225
13. Hadwin, A., Nesbit, J., Jamieson-Noel, D., Code, J., Winne, P.: Examining trace data to explore self-regulated learning. Metacognition Learning 2 (2007) 107--124
14. Fill Giordano, R., Litzenberger, M., Berthold, M.: On the assessment of strategies in self-regulated learning (SRL) – Differences in adolescents of different age group and school type. Poster at 9. ÖGP Tagung, Salzburg (2010)

Managing Ethical Thinking

Mayya Sharipova, Gordon McCalla

ARIES Lab, Dept. of Computer Science, University of Saskatchewan
{m.sharipova,gordon.mccalla}@usask.ca

Abstract. The main set of reasoning tools needed for the Professional Ethics domain is metacognitive. Students need to be able not only to analyze case studies, commonly used in this kind of domain, but also be able to analyze their own analysis. We have developed a tool called Umka to implicitly support students in evaluating and regulating their ethical analysis. An experiment was carried out where computer science students studying professional ethics used Umka. Results of this experiment are shown, and further steps are discussed on how to make Umka's metacognitive support more explicit.

Keywords: ethical thinking, metacognition, case analysis

1 Introduction

Metacognition is defined as the ability to be aware of, monitor, and evaluate one's own thinking. In the context of Professional Ethics this translates into the learner's ability to be aware of, evaluate and, if necessary, regulate his or her own ethical thinking. Professional Ethics is commonly taught through the analysis of case studies, which present certain professional issues and dilemmas. Students are asked to provide solutions to resolve these dilemmas, and supply justifications for their judgment. The reasoning behind these justifications is a big part of what constitutes "ethical thinking".

Ethical thinking by itself involves many metacognitive activities such as recognizing the complexities of your circumstances, anticipating the consequences of actions, considering the effect of actions on others, the critical appraisal of message source, quality of appeal etc. The foundation researcher in metacognition Flavell [1] considered these activities to be metacognitive in nature, and important for making wise and thoughtful life decisions.

But besides these activities students also need to be evaluate and regulate their ethical thinking. Students have to be able to analyze their own arguments and motivations, to make sure they have covered all the facts, have not factored in their own beliefs or prejudices too strongly, have uncovered all the possible directions for analyzing the case, and have weighed their arguments against one another well in reaching their conclusion. Students need to have skills to articulately and consistently justify their moral judgements, skills for analysis and critique of others' and their own convictions, and skills for forming their own convictions. Developing all these skills in students are important goals of ethics education [2].

Several systems have been developed to support students in structuring their ethics case analysis. These systems walk students through the steps of ethical analysis by providing instructions and asking students to fill in predefined forms. Examples of such systems are Ethos [3] and the PETE system [4]. We have not found systems that support students beyond structuring their ethical analysis, and in particular there doesn't seem to be support for students learning the more complex processes of evaluating and regulating ethical analysis.

2 Umka as a Tool for Evaluating and Regulating Ethical Thinking

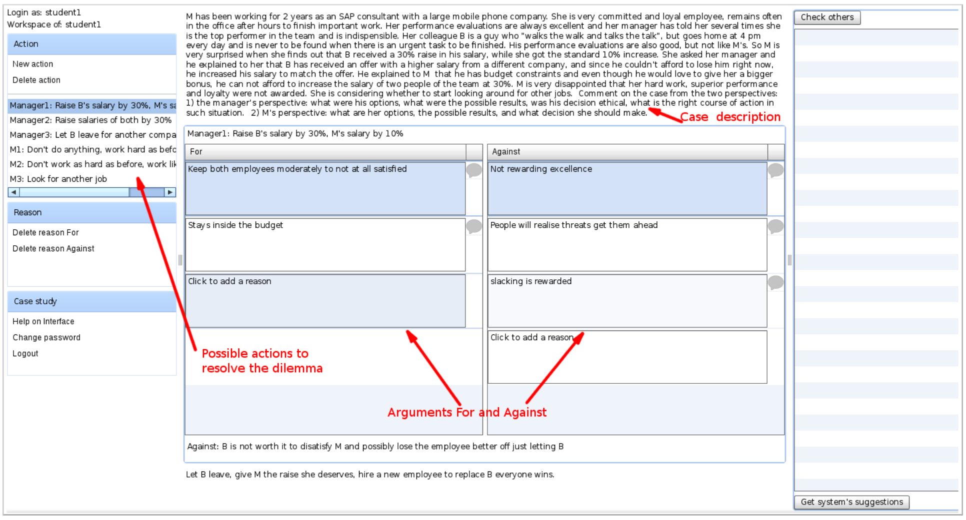
We have developed a computer tool Umka (screenshot in Figure 1) where students analyze a given case study both individually and through collaboration with one another by seeing each others' analyses and commenting on each others' arguments.

Umka also invites students to cognitively monitor their own ethical analysis, and adopt strategies for its improvement. This is done in Umka implicitly through an open group learner model of students' analysis. Bull and Kay [5] suggest that there is "potential to support metacognitive activity in a less explicit manner" though open learner models. And an important question that these researchers raise is "how to design and present a learner model that can best support reflection and particularly how to do it in ways that facilitate learning of the domain and of metacognitive skills".

If we consider the ethics domain, domain knowledge here is the formed convictions on important professional issues. Metacognitive skills are skills for evaluating one's own convictions, and strategies to form them such as looking at the issue from various points of view, exposure to the opinions of others, criticizing your own and others' convictions, overcoming criticism, or changing your convictions in response to the criticism.

The open learner model in Umka reflects how well-formed are learners' convictions or positions. The well-formedness of a learner position is determined by how broad it is in terms of different reasons the learner considered, and how well-argued it is in terms of how much the learner was able to persuade others in his or her reasoning. We have adopted the circle visualization for this (Figure 2). The size of the circle reflects the breadth of the student's position, which is determined by the number of different arguments the student has for and against a particular action in a case study. The darkness of the circle reflects the well-formedness of the student's position. The more the arguments and comments of the student are accepted by others, the more well-formed is the student's position, and the darker is the student's circle. [6] has more details on how the visualization is computed.

We expected that our open group learner model will trigger students to cognitively evaluate their convictions and adopt strategies for forming their convic-



Login as: student1
Workspace of: student1

Manager1: Raise B's salary by 30%, M's sa
Manager2: Raise salaries of both by 30%
Manager3: Let B leave for another compa
M1: Don't do anything, work hard as befo
M2: Don't work as hard as before, work lik
M3: Look for another job

M has been working for 2 years as an SAP consultant with a large mobile phone company. She is very committed and loyal employee, remains often in the office after hours to finish important work. Her performance evaluations are always excellent and her manager has told her several times she is the top performer in the team and is indispensible. Her colleague B is a guy who "walks the walk and talks the talk", but goes home at 4 pm every day and is never to be found when there is an urgent task to be finished. His performance evaluations are also good, but not like M's. So M is very surprised when she finds out that B received a 30% raise in his salary, while she got the standard 10% increase. She asked her manager and he explained to her that B has received an offer with a higher salary from a different company, and since he couldn't afford to lose him right now, he increased his salary to match the offer. He explained to M that he has budget constraints and even though he would love to give her a bigger bonus, he can not afford to increase the salary of two people of the team at 30%. M is very disappointed that her hard work, superior performance and loyalty were not awarded. She is considering whether to start looking around for other jobs. Comment on the case from the two perspectives: 1) the manager's perspective: what were his options, what were the possible results, was his decision ethical, what is the right course of action in such situation. 2) M's perspective: what are her options, the possible results, and what decision she should make.

Case description

Manager1: Raise B's salary by 30%, M's salary by 10%

For	Against
Keep both employees moderately to not at all satisfied	Not rewarding excellence
Stays inside the budget	People will realise threats get them ahead
Click to add a reason	slacking is rewarded
	Click to add a reason

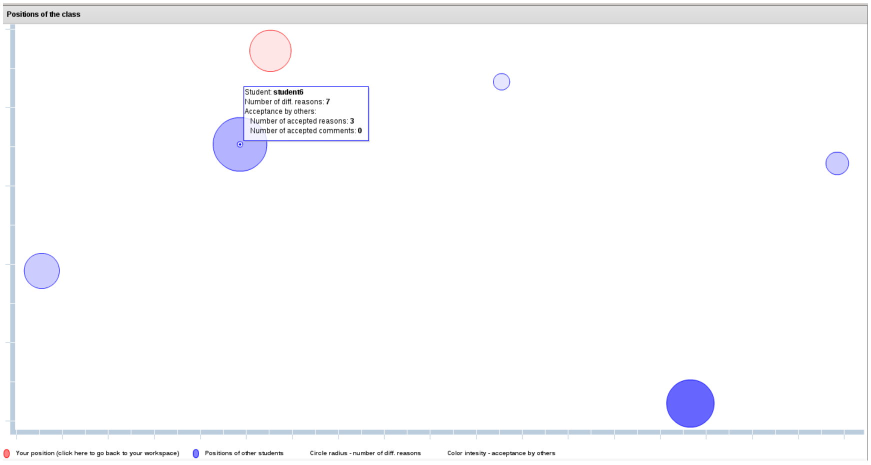
Possible actions to resolve the dilemma

Arguments For and Against

Against: B is not worth it to disatisfy M and possibly lose the employee better off just letting B

Let B leave, give M the raise she deserves, hire a new employee to replace B everyone wins.

Fig. 1. A screenshot of the Umka system. Once logged in a student sees the case description in the top middle part, and possible actions to resolve the case dilemma in the left part. The student puts his/her arguments for and against every action in the middle.



Positions of the class
Your position (click here to go back to your workspace)
Positions of other students
Circle radius - number of diff. reasons
Color intesity - acceptance by others

Student: **student6**
Number of diff. reasons: **7**
Acceptance by others:
Number of accepted reasons: **3**
Number of accepted comments: **0**

Fig. 2. Umka's visualization. A student sees his/her position as a red circle, and positions of others - as blue circles. The distance between the circles reflects the semantic distance between the corresponding positions.

tions. Our experiment described in the next section was designed to evaluate how effective was the proposed learner modeling in stimulating positive metacognitive behaviors in students, and how much students' own evaluation of their positions corresponds with the evaluation of their positions in our learner model.

3 Experiment and Results

In our two previous studies [6] we investigated the effect of Umka's support on students' behavior and the quality of students' analysis, and evaluated the accuracy of the learner modelling. The specific goal of our third experiment was more qualitative than the other two, essentially to probe more deeply into the effect of Umka on the cognition and metacognition of the students. In the third experiment we used the Umka tool for one of the assignments in an undergraduate course called "Ethics in Computer Science" at the University of Saskatchewan. Six students taking this class were analyzing a case study in the Umka tool concerning issues that may arise in the workplace. With only six students, the experiment is, of course, at best illuminative, not definitive, and there was no point in doing statistical analysis.

We were interested what students will do when they see their own learner models, and learner models of their classmates. The open learner model in Umka provoked in students certain behaviors for regulating their ethical thinking. After seeing the visualization of their learner models, students visited analyses of other students, commenting on the arguments of others, and revisited their personal analyses by adding more arguments into them. Thus, 54% of all students' arguments are arguments that have been added after seeing the visualization or analyses of other students. 55% of these added arguments were found to be good arguments by the instructor. All students except one were visiting analyses of others, and all students except one added new arguments after seeing their learner models or analyses of other students. There were 12 comments of the students on each others' arguments.

We compared these results with the results from the Wiki system that the students used for ethical analysis of another case study before they used the Umka system. In comparison, in the Wiki system the students didn't exchange any comments with each other, and the students didn't revise their own arguments.

In the post-study questionnaire we asked students to evaluate their ethical thinking and compare it with the Umka visualization, specifically asking how much the visualization was able to reflect the breadth and well-formedness of their positions. Unfortunately only one student out of six filled in the questionnaire. This student stated that the visualization didn't reflect much about his position because as he said ".. I feel that my 2 reasons were more detailed then 5 one sentance [sic] details that other students gave. Although if they expanded their reasons more I feel I would try [to] increase my position".

4 Conclusion and Future Directions

One of the goals behind Umka's development was to support students in managing their ethical analysis. This support is organized implicitly through Umka's interface, visual feedback on the breadth and depth of students' arguments, and encouragement to look at others' arguments. While our study was a small one, making definitive conclusions premature, the results were positive. Using Umka, students were motivated to actually argue and discuss with one another and to examine their own arguments; they were able to regulate their ethical analysis. There was not enough data to judge how well students were able to evaluate their ethical thinking and the degree they agreed with Umka's evaluation. A possible future direction is to organize Umka's visualization as an open negotiated learner model [7] to further stimulate metacognitive behaviors in students. Another possible direction is the introduction of explicit learner centered system suggestions on structuring and regulating ethical case analysis.

Metacognition plays an important role in learning Professional Ethics. The ability not just to analyze a case, but to analyze the analysis is fundamental to the ethics domain. Thus, the ethics domain is a perfect domain to explore metacognition, and further research is required to understand how it can be best supported by a computer environment.

Acknowledgements. The authors wish to thank the Natural Sciences and Engineering Research Council of Canada for their funding of this research project.

References

1. Flavell, J. H.: Metacognition and cognitive monitoring. American psychologist, 34(10), 906–911 (1979)
2. The Hasting Center: The teaching of ethics in higher education. Tech. rep., The Hastings Center, Institute of Society, Ethics and the Life Sciences, Hastings-on-Hudson, N.Y. (1980)
3. Searing, D.R.: Harps ethical analysis methodology: Method description, version 2.0.0. Technical report, Taknosys Software Corporation, (1998)
4. Goldin, I.M., Ashley, K.D., Pinkus, R.L.: Introducing PETE: computer support for teaching ethics. In Proceedings of the 8th International Conference on Artificial Intelligence and Law, 94–98. ACM, (2001)
5. Bull, S., Kay, J.: Metacognition and Open Learner Models, in I. Roll and & V. Aleven (eds), Proceeding of Workshop on Metacognition and Self-Regulated Learning in Educational Technologies, International Conference on Intelligent Tutoring Systems, 7–20 (2008)
6. Sharipova, M., McCalla, G.: Modelling Students Knowledge of Ethics, To appear in Proceedings of the 2013 conference on Artificial Intelligence in Education (2013)
7. Kerly, A., Hall, P., Bull, S.: Bringing chatbots into education: Towards natural language negotiation of open learner models. J. of Knowledge-Based Systems, special issue from 26th SGAI Conference on Innovative Techniques and Applications of Artificial Intelligence (AI 2006), 20, 2, Elsevier, 177-185 (2007)

A Framework for Self-Regulated Learning of Domain-Specific Concepts

Bowen Hui

Department of Computer Science, University of British Columbia Okanagan
and
Beyond the Cube Consulting Services Inc.

Abstract. Research in self-regulated learning environments has focused on student motivation, development of metacognitive skills, learning strategies, and individual differences. Equally important is the modeling of domain-specific concepts and the ability for students to learn them under their preferred environment. In this paper, we present a general framework for modeling domain-specific concepts that support self-regulated learning across different domains. Our framework is motivated by a well-established pedagogical tool called the *concept map*.

Keywords: Concept map, self-regulated learning, individualized learning paths, performance monitoring, relevance perception

1 Introduction

One of the most important factors in course design is the development of a *concept map* [1], which is the overall picture of the relationship between the course concepts and the learning elements. As educators, we are often concerned with student performance regarding specific concepts and learning outcomes, and whether they understand the connections among the various course components. While we design assessments to help students achieve various learning outcomes, the interconnectedness of the concepts assessed in course activities make it hard for us to tease apart what students excel in and what they find difficult. In order to better help the students, ideally, educators should be able to point to an assessment piece, see the corresponding performance level, and know immediately which concepts students have trouble with and which learning outcomes may be in jeopardy. Likewise, students should have access to metrics about their own progress so that they can monitor and shape their own learning process. Much like the benefits that project management software offer to managers and employees, we wish to deliver analogous information in the context of a course that lets students and instructors manage the learning process. As such, we argue that an online course tool is needed to overcome these challenges by visually presenting key concepts and their connections to other elements. We present a general framework called the *Concept Navigator* for just this purpose. While its design is motivated by the needs of educators, this framework also supports students in a self-regulated learning environment. We believe that the

Concept Navigator will empower both students and educators by providing them with an explicit view of student progress with respect to a course concept map and the expected learning outcomes.

2 The Concept Navigator Framework

As new educational paradigms, such as flexible learning and flipped classrooms, become mainstream, there is a growing need to have the proper tools in place to support methods of student-initiated and student-directed learning [2]. The Concept Navigator is a general framework for visualizing course concepts, their relationships to each other, as well as their relationships to other course elements such as learning outcomes and assessment pieces. The backbone of this framework is driven by a course concept map, as concept mapping has been shown to support self-directed, experimental, and networked learning (see [2] for details). Although the concept map has long been available to educators for course design purposes, in our experience, most instructors do not use it in designing courses or in articulating the roadmap of a course to students. From a pedagogical standpoint, we believe that the development of a concept map is crucial to the successful delivery of a course. For this reason, our framework is designed to have instructor-defined concept maps of courses, rather than data-driven [3] or editable concept maps of learners [4] as proposed by alternative approaches.

The concept map alone is simply a set of concepts and their relationships. In our framework, we model additional entities and relationships as depicted in Figure 1. For example, a concept is associated with many learning outcomes, and can be included in an activity (e.g., reading) or exercised in a question (which belongs to either an assignment or a quiz). Also, note that a learning outcome is related to other learning outcomes because some outcomes may serve as prerequisite skills. Finally, a profession (e.g., Programmer, System Analyst, Project Manager) may require the mastery of different sets of learning outcomes. This relationship is of particular importance because it helps students see real-world relevance of what they are learning in class.

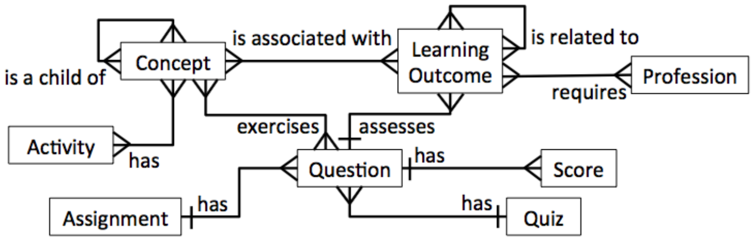


Fig. 1. The entity-relationship diagram for the Concept Navigator.

Overall, this model defines the structural content of a course from an the instructor's perspective. As such, one of our goals is to promote the use of concept maps in the process of course design. Since instructional content and style can vary, our framework is limited to supporting specific course development efforts rather than larger efforts such as degree program design (e.g., [5]). Unlike existing work in open learner models [6], we focus on the explicit communication of concepts and their interdependencies, as well as their relationships to learning outcomes and relevance to professions. Students with a good grasp of this knowledge will be able to personalize their learning experience by setting real-world driven goals and choosing their own paths based on what they want to achieve. Moreover, this framework is a concept navigation tool, without adaptive features and requiring minimal student configuration (see [7] for an alternative approach). In contrast to learning management systems such as Blackboard [8] and Moodle [9] that simply deliver course content digitally and perform simple software usage tracking, the Concept Navigator enables students to take control of their own learning process. Currently, Moodle also lets users tag course elements to learning outcomes, which is a step toward our overall design objectives.

3 A Course Prototype in the Concept Navigator

To illustrate our framework, we present a partial concept map of the course "Digital Citizenship" in Figure 2, where concepts are represented as nodes and relationships are represented as arrows. The small graphs shown on the top of the nodes indicate summary metrics of student performance, which we envision can be viewed per student or for a whole class. Student progress is implicitly shown in Figure 2 by a lack of available data in the remaining nodes.

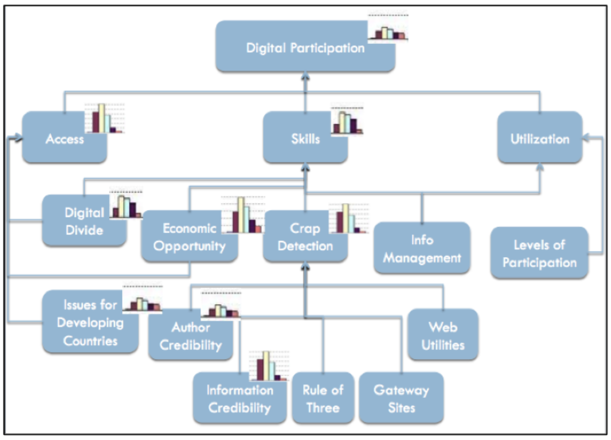
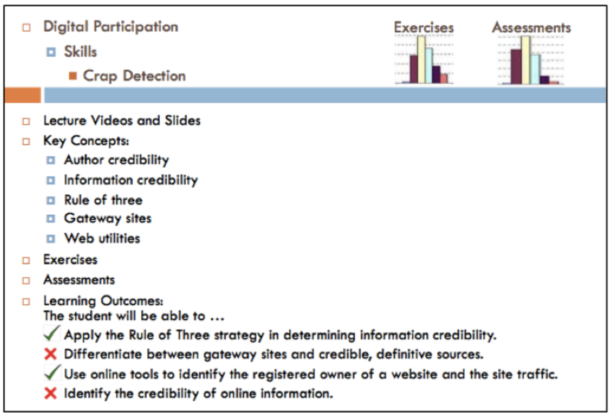


Fig. 2. A partial concept map for Digital Citizenship with summary metrics.

When a concept is selected, such as “Crap Detection”, a detailed view as in Figure 3 will be shown. Parent concepts based on Figure 2 and summary metrics are shown at the top, while related learning elements such as activities (e.g., readings, videos), questions (as part of exercises or assessments), and learning outcomes are displayed in the center. Details may be hidden or expanded.



- □ Digital Participation
  - □ Skills
    - ■ Crap Detection
- □ Lecture Videos and Slides
- □ Key Concepts:
  - □ Author credibility
  - □ Information credibility
  - □ Rule of three
  - □ Gateway sites
  - □ Web utilities
- □ Exercises
- □ Assessments
- □ Learning Outcomes:
  The student will be able to ...
  - ✓ Apply the Rule of Three strategy in determining information credibility.
  - ✗ Differentiate between gateway sites and credible, definitive sources.
  - ✓ Use online tools to identify the registered owner of a website and the site traffic.
  - ✗ Identify the credibility of online information.

Fig. 3. Detailed view of Crap Detection, showing related concepts and summary metrics at the top and hidden and expanded learning elements in the center.

Of particular interest is the display of learning outcomes which serves as a constant reminder of why certain concepts are taught as part of the course and the expectations in applying them. Moreover, Figure 3 shows a visual status for each learning outcome to indicate how likely the student has achieved a learning outcome based on the current performance levels. These statuses can be determined based on predefined thresholds or automatically learned via a history of performance data. Usability feedback will be conducted to test whether a more fine-grained visual status (e.g., a percentage) will be more appropriate than a binary status (i.e., ✓ or ✗). These metrics are helpful in providing a formative assessment so that instructors may adapt learning activities accordingly.

4 Support for Self-Regulated Learning

The Concept Navigator is designed to support students in a self-regulated learning environment. A key aspect of the concept map interface (e.g., Figure 2) is the ability for students to pursue a course in a non-linear fashion. Given a visual map of the concepts and their dependencies, students may select the concepts of interest and acquire the relevant material via an individualized learning path. The ability to see the direct connections between concepts, learning outcomes, and professions not only enables students to set goals for themselves, but it also helps to foster a positive attitude in students by knowing the importance of each learning element at hand. With the metrics associated to each concept and learning outcome, students can monitoring their own progress and, thus, increase awareness of their own educational successes and needs.

Currently, our framework assumes students take full responsibility of their own learning. Opportunities to add social and intelligent features are left for future development, such as peer information sharing forums, monitoring alerts that trigger self-reflection, and adaptive assistance to support scaffolding.

5 Future Work

We presented a framework called the Concept Navigator which supports self-regulated learning of domain-specific concepts. This framework hails students as active agents in their own learning process. We instantiated this framework with a course prototype and discussed ways to support individualized learning, goal setting, performance monitoring, reflection, and relevance perception. Our immediate next step is to design the interface for visualizing the relationships among learning outcomes and between learning outcomes and professions. Thereafter, we will create a full instance of the Concept Navigator for a specific course and test it with student users. Controlled testing to debug usability issues will be conducted prior to assessing the utility of the system by testing it in the classroom. Finally, testing in different courses will be done to validate the feasibility of this framework across multiple domains.

References

1. Novak, J., Gowin, D.: Learning How to Learn. Cambridge University Press, Cambridge MA (1984)
2. Hui, B., Crompton, C.: The need to support independent student-directed learning. In: Learning Technology for Education in Cloud, Kaohsiung, Taiwan (2013)
3. Perez-Marin, D., Alfonseca, E., Rodriguez, P., Pascual-Neito, I.: A study on the possibility of automatically estimating the confidence value of students. Journal of Computers **2**(5) (2007) 17–26
4. Mabbott, A., Bull, S.: Student preferences for editing, persuading and negotiating the open learner model. In: Intelligent Tutoring Systems, Jhongli, Taiwan (2006) 481–490
5. Gluga, R., Kay, J., Lever, T.: Foundations for modeling university curricula in terms of multiple learning goal sets. IEEE Transactions on Learning Technologies **6**(1) (2013) 25–37
6. Bull, S., Kay, J.: Open Learner Models. In: Advances in Intelligent Tutoring Systems. Springer (2010)
7. Dufresne, A.: Model of an adaptive support interface for distance learning. In: Intelligent Tutoring Systems, Montréal, Canada (2000) 334–343
8. Blackboard: `http://www.blackboard.com`
9. Moodle: `https://www.moodle.org`

Evaluation of a meta-tutor for constructing models of dynamic systems

Lishan Zhang, Winslow Burleson, Maria Elena Chavez-Echeagaray, Sylvie Girard, Javier Gonzalez-Sanchez, Yoalli Hidalgo-Pontet, Kurt VanLehn

Arizona State University, Computing, Informatics, and Decision Systems Engineering, Tempe, AZ, 85281, U.S.A.

{lishan.zhang, winslow.burleson, mchaveze, sylvie.girard, javiergs, yhidalgo, kurt.vanlehn}@asu.edu

Abstract. While modeling dynamic systems in an efficient manner is an important skill to acquire for a scientist, it is a difficult skill to acquire. A simple step-based tutoring system, called AMT, was designed to help students learn how to construct models of dynamic systems using deep modeling practices. In order to increase the frequency of deep modeling and reduce the amount of guessing/gaming, a meta-tutor coaching students to follow a deep modeling strategy was added to the original modeling tool. This paper presents the results of two experiments investigating the effectiveness of the meta-tutor when compared to the original software. The results indicate that students who studied with the meta-tutor did indeed engage more in deep modeling practices.

Keywords: meta-tutor , intelligent tutoring systems, empirical evaluation

1 Introduction

Modeling is both an important cognitive skill [1] and a potentially powerful means of learning many topics [5]. The AMT system teaches students how to construct system dynamics models. Such models are widely used in professions, often taught in universities and sometimes taught in high schools.

1.1 The modeling language, development tool and tutoring system

In our modeling language, a model is a directed graph with one type of link. Each node represents both a variable and the computation that determines the variable's value. Links represent inputs to the calculations. As in illustration, Figure 1 shows a model for the following system:

> The initial population of bacteria is 100. The number of bacteria born each hour is 10% of the population. Thus, as the population increases, the number of births increases, too. Model the system and graph the population over 20 hours.

Clicking on a node opens an editor with these tabs (and 2 others not described here):

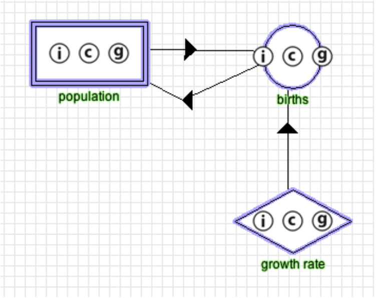


Fig. 1. A simple model.

- *Description*: The student enters a description of the quantity represented by the node.
- *Inputs*: The student selects inputs to the calculation of the node's value.
- *Calculation*: The student enters a formula for computing the node's value in terms of the inputs.

There are three types of nodes in models:

- A *fixed value* node represents a constant value that is directly specified in the problem. A fixed value node has a diamond shape, never contains incoming links, and its calculation is just a single number. For instance, "growth rate" has 0.1 as the calculation of its value.
- An *accumulator* node accumulates the values of its inputs. That is, its current value is the sum of its previous value plus or minus its inputs. An accumulator node has a rectangular shape and always has at least one incoming link. For instance, the calculation tab of "population" states that its initial value is 100 and its next value is its current value + births.
- A *function* node's value is an algebraic function of its inputs. A function node has a circular shape and at least one incoming link. For instance, "births" has as its calculation "population * growth rate."

The students' task is to develop a model that represents a system described by a short text. They can create, edit and delete nodes using the node editor. When all the nodes have calculations, students can click the Run Model button, which performs calculations and draws graphs of each nodes' values over time. The system described so far is just a model development tool.

AMT has a simple tutoring capability. Each tab of the node editor has a *Check* button which turns its fields red if they are incorrect and green if they are correct. Each tab also has a *Give up* button that fills out the tab correctly. Thus, the system described so far is just a simple step-based tutoring system with minimal feedback on demand and only one kind of hint: a bottom-out hint.

1.2 The meta-tutor

Unfortunately, it is a rare for students to think semantically in terms of what the nodes, inputs and calculations mean actually mean. Students prefer to think of model elements syntactically, like puzzle pieces that need to be fit together. This shows up in a variety of ways, including rapid guessing, nonsensical constructions and the use of syntactic rather than semantic language to refer to model elements. The literature on model construction (reviewed in [5]) sometimes refers to these two extremes as Deep vs. Shallow modeling. The objective of the AMT system is to increase the relative frequency of Deep modeling.

A variety of methods for increasing the frequency of Deep modeling have been tried [5]. For instance, nodes can bear pictures of the quantities they represent, or students can be required to type explanations for their calculations. One of the most promising methods is *procedural scaffolding*, wherein students are temporarily required to follow a procedure; the requirement is removed as they become competent. This technique was used by Pyrenees [2], where it caused large effect sizes.

We adapted Pyrenees' procedure to our modeling language and called it the Target Node Strategy. The strategy requires students to focus on one node, called the target node, and completely define it before working on any other node. This decomposes the whole modeling problem into a series of atomic modeling problems, one per node. The atomic modeling problem is this: Given a quantity, find a simple calculation that will compute its values in terms of other quantities without worrying about how those other quantities values will be calculated. This is a much smaller problem than the overall challenge of seeing how the overall model can be constructed.

As an illustration, let us continue the bacteria population example and suppose that the target node is "number of bacteria born per hour." The ideal student might think:

> "It says births are 10% of the population, so if I knew population, then I could figure out the number of births. In fact, I could define a node to hold the 10%, and then the calculation would multiply it and population. But do I need initial population or current population? Oh. The number of bacteria born is increasing, so I must need current population, because it is also increasing."

This is one form of deep modeling. By requiring students to finish one node before working on another, the Target Variable Strategy encourages students to examine the system description closely because it is the only resource that provides relevant information. When they are allowed to work on any tab on any node, then they jump around trying to find a tab that can be easily filled in. This is a common form of shallow modeling, and the Target Node Strategy discourages it.

In addition to requiring the students to follow the Target Node Strategy, the meta-tutor nags students to avoid guessing and abuse of the Give Up button, just as the Help-Tutor [3] did. Because neither the strategy nor the advice on help seeking are specific to the domain (e.g., population dynamics), we consider them to be meta-cognitive instruction.

2 Evaluation

2.1 Experiment Design

The experiment was designed as a between-subject single treatment experiment with a control condition, where the meta-tutor was off, and an experiment condition, where the meta-tutor was on. The difference between the conditions occurred only during a training phase where students learned how to solve model construction problems. In order to assess how much students learned, a transfer phase followed the training phase. During the transfer phase, all students solved model construction problems with almost no help: the meta-tutor, the Check button and the Give-up button were all turned off, except in the Description tab where the Check button remained enabled to facilitate grounding. Because system dynamics is rarely taught in high school, no pretest was included in the procedure. We conducted two experiments with 44 students participating in the first experiment and 34 students in the second experiment.

2.2 Hypotheses and Measures

Hypothesis 1 is that the meta-tutored students will use deep modeling more frequently than the control students during the *transfer* phase. We used the three measures below to assess it.

- The *number of the Run Model button presses* per problem.
- The *number of extra nodes* created, where extra nodes are defined as the nodes that can be legally created for the problem but are not required for solving the problem.
- The *number of problems completed* during the 30 minute transfer period.

Hypothesis 2 is that meta-tutored students will use deep modeling more frequently than the control group students during the *training* phase. The three dependent measures used to evaluate this hypothesis are described below:

- *Help button usage:* was calculated as $(n_{wc}+3n_{gu})/n_{rn}$, where n_{wc} is the number of Check button presses that yielded red, n_{gu} is the number of Give-up button presses, and n_{rn} is the number of nodes required by the problem.
- *The percentage of times the first Check was correct.*
- *Training efficiency:* was calculated as $3n_{cn} - n_{gu}$ where n_{cn} is the number of nodes the student completed correctly ($3n_{cn}$ is the number of tabs), and n_{gu} is the number of Give-up buttons presses.

Hypothesis 3 is that the experimental group students, who were required to follow the Target Node Strategy during training, would seldom use it during the transfer phase. To evaluate this hypothesis, we calculated the proportion of student steps consistent with the target node strategy.

2.3 Results

Table 1 summarizes the results of experiment 1 and experiment 2.

3 Conclusion and future work

Although we achieved some success in encouraging students to engage in deep modeling, there is much room for improvement. If the meta-tutor had been a complete success at teaching deep modeling, we would expect to see students supported by the meta-tutor working faster than the control students. The stage is now set for the last phase of our project, where we add an affective agent to the system [4], in order to encourage engagement and more frequent deep modeling.

Measure (predicted dir.)	***Experiment 1 (N=44)***	***Experiment 2 (N=33)***
Transfer phase (Hypothesis 1)		
Run model button usage (E<C)	E<C (p=0.31, d=0.32)	E≈C (p=0.98, d=-0.0093)
Extra nodes (E<C)	**E<C (p=0.02, d=0.80)**	E<C (p=0.47, d=0.26)
Probs completed (E>C)	E≈C (p=0.65, d=0.04)	E<C (p=0.09, d=−0.57)
Training phase (Hypothesis 2)		
Help button usage (E<C)	**E<C (p=0.04, d=0.68)**	**E<C (p=0.02, d=0.89)**
Correct on 1st Check (E>C)	Missing data	**E>C (p=0.015, d=0.98)**
Efficiency (E>C)	E<C (p=0.05, d=−0.70)	E>C (p=0.59, d=0.19)
Transfer phase use of Target Node Strategy (Hypothesis 3)		
Usage (E=C)	Missing data	E≈C (p=0.59, d=−0.19).

Table 1. Results of Experiment 1 and 2: E stands for the meta-tutor group, and C stands for the control group. Reliable results are bold.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 0910221.

References:

1. CCSSO.: The Common Core State Standards for Mathematics, Downloaded from www.corestandards.org on October 31 (2011)
2. Chi, Min, & VanLehn, K.: Meta-cognitive strategy instruction in intelligent tutoring systems: How, when and why. Journal of Educational Technology and Society, 13(1). 25-39 (2010)
3. Roll, I., Aleven, V., McLaren, Bruce, Ryu, Eunjeong, Baker, R.S.J.d., & Koedinger, K. R.: The Help Tutor: Does metacognitive feedback improve student's help-seeking actions, skills and learning. In M. Ikeda, K. Ashley & T.-W. Chan (Eds.), Intelligent Tutoring Systems: 8th International Conference, pp. 360-369. Berlin: Springer (2006)
4. Girard, S., Chavez-Echeagaray, M. E., Gonzalez-Sanchez, J., Hidalgo-Pontet, Y., Zhange, L., Burleson, W. & VanLehn, K.: Defining the behavior of an affective learning companion in the Affective Meta-Tutor project. In Proceedings of AI in Education (2013)
5. Treagust, David F., Chittleborough, Gail, & Mamiala, Thapelo.: Students' understanding of the role of scientific models in learning science. International Journal of Science Education, 24(4), 357-368 (2002)
6. VanLehn, K. (in press). Model construction as a learning activity: A design space and review. *Interactive Learning Environments*.