Intelligent Support in Exploratory and Open-ended Learning Environments

Learning Analytics for Project Based and Experiential Learning Scenarios

held in conjunction with

Seventeenth International Conference on
Artificial Intelligence in Education (AIED 2015)

Monday, June 22, 2015
Madrid, Spain

Workshop Co-Chairs:

Manolis Mavrikis[1], Gautam Biswas[2], Sergio Gutierrez-Santos[3],
Toby Dragon[4], Rose Luckin[1], Daniel Spikol[5], James Segedy[2]

[1]London Knowledge Lab, University College, London
[2]Electrical Engineering and Computer Science, Vanderbilt University
[3]London Knowledge Lab, School of Computer Science, Birkbeck
[4]Department of Computer Science, Ithaca College, US
[5]Faculty of Technology and Society, Malmo University

http://link.lkl.ac.uk/iseole15
https://aied15learninganalytics.wordpress.com

Table of Contents

Preface

By encouraging interaction, exploration and experimentation in environments that directly represent the domain to the learner, Exploratory Learning Environments (ELE) adhere to constructivist theories of learning that emphasize learners' control to construct their own understanding. More generally, Open-ended Learning Environments (OLEs) offer students opportunities to take part in authentic and complex problem-solving and inquiry learning activities. These environments provide learning context and a set of tools to support learners while they engage in many activities, including (i) seeking and acquiring knowledge and information, (ii) applying that information to a problem-solving context, (iii) assessing the quality of the constructed solution, (iv) evaluating and reflecting on the overall approach, and (v) assessing and enacting cognitive and metacognitive processes.

However, there are several factors that prevent appropriate learning within ELEs or OLEs. The structure of the activity sequences and the level of support by teachers, peers, technologies are crucial determinants of learning. This is particularly true in domains where knowledge is not a directly observable outcome of a situation under exploration (e.g. simulators) but is externalized by cognitive tools in the environment. There is a wealth of learning sciences literature about support for learning in exploratory environments, but developing the technology to support these still faces several impressive challenges that the community is only beginning to address.

At the same time the migration of technology from the desktop to the wider learning environment provides the opportunity to collect data about learners' interactions with a greater bandwidth of learning resources. Smart phones, tablets and technologies embedded in the fabric of the environment are now commonplace in educational settings. In parallel with these developments, there has been great progress in developing techniques to analyse learning interactions through the large amount of data that is generated by these various systems. This kind of learning analytics offers the potential for novel feedback and scaffolding to support project-based and experiential learning that involves physical computing projects and other hands-on type projects.

The papers submitted to this workshop address various aspects of the above-listed issues, which are all at the heart of the AIED community's interest.

Summarizing the papers in brief, Chase et al. and Mazziotti et al. focus mostly on the *design and evaluation of exploratory learning environments*. Chase et al. in particular describe the design of an ELE to support invention activities, inspired by a model of naturalistic teacher guidance. Mazziotti et al. present a pedagogical intervention model that selects and sequences exploratory learning activities and structured practice activities. Four papers focus more on the *tools, algorithms and approaches* behind the implementation of intelligent support in ELEs. Karkalas et al. evaluate requirements and present a prototype for learning analytics for constructionist mathematical e-books. Segedy and Biswas use coherence analysis to provide measures of the quality of students' problem-solving processes. Silva et al. propose an automatic rating system to assess students and to sequence activities. Harpstead et al. demonstrate a method of accelerating model development for both knowledge and skills by applying a concept formation algorithm.

Lastly, two papers focus specifically on *Learning analytics for project based and experiential learning scenarios.* Luckin et al. present an analysis framework for project-based learning situations that involve the use of technology. Spikol et al. present the design of a visual-based programming language for physical computing and mobile tools to invite learners to actively document and reflect on their projects in a way that creates possibilities of intelligent support and learning analytics.

This workshop builds on the previous work from several editions of the Intelligent Support in Exploratory Environments workshop, and the Scaffolding in Open-Ended Learning Environments in AIED 2013. The format of the workshop is based on a question-oriented organisation around open problems raised by the papers accepted for the workshop. It also includes a posters and hands-on interactive session for participants to present prototypes and get or provide feedback. Our website (http://link.lkl.ac.uk/iseole15) provides more information as well as the current and previous proceedings.

Manolis Mavrikis, Gautam Biswas, Sergio Gutierrez-Santos, Toby Dragon, Rose Luckin, Daniel Spikol, James Seged

Workshop Co-Chairs

The design of an exploratory learning environment to support Invention

Catherine C. Chase, Jenna Marks, Deena Bernett, and Vincent Aleven

[1]Teachers College, Columbia University, New York, United States
(chase, jnm2146, dlb2175) @tc.columbia.edu
[2]Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, United States
aleven@cs.cmu.edu

Abstract. We describe the design of the Invention Coach, an intelligent, exploratory learning environment (ELE) to support Invention, an exploratory learning activity. Our design is based on a two-pronged approach. Our own study of naturalistic teacher guidance for paper-based Invention uncovered phases in the Invention process. Prior research on the mechanisms of learning with Invention activities revealed specific instructional strategies. These two sources informed the design of the guidance offered by the Invention Coach. To our knowledge, this is the first design of a guided environment for Invention activities inspired by a model of naturalistic teacher guidance. Our work offers insight into styles of guidance that could apply to other exploratory learning environments.

Keywords: intelligent learning environment, human tutoring, exploratory learning, intelligent tutors

1 Introduction

While exploratory tasks support the constructivist nature of learning and have the potential to enhance 21st century skills, there is broad agreement that learners need guidance in their exploration [1]. But what kind of guidance will help learners to engage in productive exploration without eliminating the exploratory nature of the task? Designers of exploratory learning environments have investigated this question through various lenses – types of learner feedback [2, 3], "cognitive tools" for inquiry [4], and participation structures [5]. We explore the question of effective guidance for exploration in the context of an exploratory learning task called Invention, where learners invent their own formulas to describe scientific phenomena. We are now in the process of developing an intelligent, exploratory learning environment (ELE) called the Invention Coach, which scaffolds students through the Invention process.

Invention is an exploratory task that invites students to engage with deep, conceptual ideas by analyzing a set of data [6]. Students are asked to invent an expression of an underlying structure that runs throughout a set of contrasting cases. Cases are examples of phenomena with predesigned contrasts that highlight key features, providing students with clues to the abstract, underlying concepts. After exploring the cases and inventing their own structures, students are told the canonical structures, through traditional expositions (lecture, reading). Prior work suggests that Invention creates "a time for telling," preparing students to appreciate the "mathematical work" of equations [6] or "function of tools for solving relevant problems" [7].

Figure 1 shows an Invention task our computerized Invention Coach is designed to support. In this "Crowded Clowns" task, students are asked to invent a numerical "index" to describe how crowded the clowns are in each set of buses. Though students do not realize it, they are inventing the equation for density (d=m/v, where density is the number of objects crowded into a space). Most students initially attempt to describe crowdedness using a single feature – the number of clowns. They do not realize that crowdedness must consider two features related in a *ratio* structure (e.g. #clowns ÷ #boxes). The six buses in Figure 1 are contrasting cases designed to highlight the critical features of "crowdedness." For example, by contrasting cases A1 and B1 (see Figure 1), which both have 3 clowns but different-sized buses, students may notice that clowns alone cannot account for crowdedness, and space must be considered as well. Through an iterative process of generating and evaluating their inventions, students begin to realize that a workable solution must involve both features in some kind of relational structure. While many students do not produce the correct formula, the invention process prepares them to learn from a later lecture on ratio structures, which is the targeted content of our instruction.

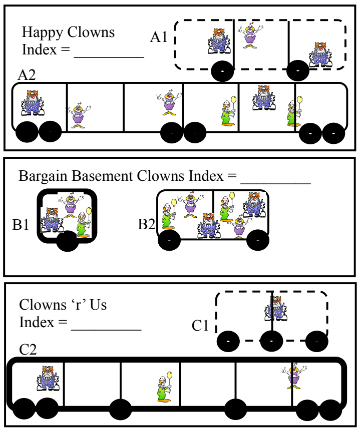

Fig 1. Invention task, adapted from Schwartz et al., 2011.

Invention activities are very successful in supporting transfer. In several studies, Invention has been more effective than traditional instruction at enhancing transfer and deep learning in science and math domains, both with adolescents and adults [6, 8, 9, 10]. But in most studies, students need subtle guidance from a teacher to engage in productive invention. In a move towards scaling up, we are developing a computer-based Invention Coach that will ultimately provide adaptive guidance as students engage in Invention. Through the design of the Invention Coach, we also explore what types of guidance are most effective in scaffolding an exploratory task.The most applicable related work comes from Roll, Aleven, and Koedinger [11], who developed an ELE for Invention activities in statistics. The learning environment we propose will share some characteristics with their Invention Lab but will differ in a fundamental way. While Roll et al.'s technology was developed through rational analysis of the task and empirical study of components of the Invention process, our Invention Coach is modeled on guidance from a human teacher.

To develop the Invention Coach, we are following a multi-phase approach of formal empirical research interspersed with design cycles and informal user testing. We began with a study of naturalistic human teachers' guidance of Invention and a review of the literature on learning with Invention. In the following section, we briefly review the results of both. We then describe the design of our current Invention Coach, focusing on the pedagogical elements of our design rather than the technical aspects underlying it. We are now in the process of implementing a Wizard-of-Oz version of the Coach, though we plan to build a fully adaptive system in the future.

2 A Two-pronged Approach to Design

The design of the Invention Coach was driven by a combination of our own empirical work and prior research and theory on Invention. Our study of naturalistic teacher guidance demonstrated the process of Invention by explicating the various subgoals teacher-student pairs tackle as they work towards a solution. The specific instructional strategies embedded in our Coach were drawn from research and theories on the mechanisms that make Invention a successful instructional paradigm.

Our analysis of naturalistic teacher guidance uncovered a process model of guided Invention with four phases [12]. In the "understand the problem" phase, teachers explained the task goal and constraints to students who were confused by the ill-defined goal of inventing an "index." In the "notice features" phase, teachers guided students to notice key features they often overlooked (most often bus size) or to think conceptually about what "crowdedness" means. In the "produce and reflect on an Invention" phase, students generated their numerical index and teachers helped them evaluate whether it was correct. There was also a "math calculation" phase, in which teachers and students worked to simplify and manipulate fractions or count key features. Informally, we noted that phases were not completed in a linear fashion; teacher-student pairs moved back-and-forth between them. As a result, our initial prototype Invention Coach supports each phase, without prescribing a specific phase order.

While the study of naturalistic tutor guidance revealed the subgoals of solving an Invention problem, specific instructional strategies were derived largely from the existing literature on Invention. Instructional strategies were designed to scaffold three core components of the Invention paradigm: noticing deep features of a domain, monitoring errors, and withholding direct feedback. First, noticing deep features of a domain is a critical step for problem-solving success. For instance, novices often focus on the surface features of a problem while experts focus on the deep principles that underlie a problem solution [13]. An effective way to help novice learners notice key features is to have them compare and contrast example cases that explicate the features [7]. Our carefully designed contrasting cases systematically differ on key features, so that certain pair-wise comparisons reveal the necessity of considering a not-so-obvious feature. Second, Invention helps learners to identify gaps in their understanding, which they can then seek to fill in later expository instruction [14]. Through the process of monitoring and reflecting on their solution attempts, learners often come to see that their invention is inadequate. When they later receive a lecture on the canonical problem solution, they are prepared to understand how it avoids the errors they made in their own solution attempts. We scaffold monitoring by encouraging learners to explain their solutions. Related work on self-explanation suggests that it strongly enhances metacognitive monitoring [15]. A third critical component of Invention is that giving away the answer or showing students how to solve the problem cuts off learners' exploration and hinders their ability to notice and monitor [9]. Thus, instead of providing direct right/wrong feedback and elaborative explanatory feedback, our system exposes inconsistencies in the learner's solution. In sum, the three instructional strategies our system employs are (1) encouraging learners to contrast cases (2) inviting learners to explain their solutions and (3) providing feedback that exposes inconsistencies in a learner's solution.

3 Design of Invention Coach Prototype

Our research findings along with prior work on Invention informed the design of the Invention Coach. We designed instructional components corresponding to each phase of the Invention process model derived from our study. Additionally, some components scaffold students as they engage in the core learning mechanisms of the Invention paradigm. Our initial prototype was designed to be operated by a "Wizard-of-Oz" (the experimenter), who can launch the student into instructional components in any order, based on her assessment of the student's current knowledge state. While we ultimately plan to build a fully adaptive Invention Coach, the Oz configuration allows for flexible application of process phases across students. Perhaps more importantly, the Oz configuration will help us identify the trigger conditions for each type of coach guidance. We are now in the throes of building our first prototype Invention Coach. We are using the Cognitive Tutor Authoring Tools (CTAT, [16]) to build our ILE as an example-tracing tutor with additional custom programming.

In our Invention Coach, the student is initially left to work independently on his invention. During this independent work time, students typically inspect the cases provided and begin entering potential index numbers for each case. Students can also click the "rules tab" to re-read the rules that their index must follow, the "calculator tab" to display an on-screen calculator, the "notepad" tab to display an on-screen notepad, or the "help" or "submit" buttons to request feedback from Oz. Oz only provides guidance in response to the student's request for feedback, or whenever the student has been working uninterrupted for five minutes.

There are two types of guidance that Oz can provide: modules and hints. A module is a short exchange between the computer and student focused on a particular subgoal.

For example, our "ranking module" (Figure 2A) asks students to rank the bus companies from most to least crowded. After the student ranks the companies, the system automatically provides feedback and, if needed, additional scaffolding. Once the student has successfully ranked the companies, the module ends, and the student is left to work independently again. Hints represent the second type of guidance Oz can provide. Hints are much simpler than modules, consisting of a single text bubble displayed to the student. The system provides largely high-level hints with broad suggestions and never gives a "bottom-out" hint, which would give away the answer.

Each of the instructional components included in the Invention Coach was designed to guide students through one of the four process phases revealed in our analysis of teacher guidance (Table 1). Most components employ one of three instructional strategies that support the mechanisms of learning with Invention: encouraging students to contrast cases, inviting students to explain their solutions, and provide feedback that exposes inconsistencies in the student's inventions.

Table 1. Invention Process Model, Instructional Strategies, and Instructional Components

Process Phases	Process Description	Instructional Strategy	Instructional Component
Understand the Problem	Explain or describe task goal and constraints	Expose inconsistencies	Rule-related hints Rules tab
Notice Features	Notice key features of the underlying structure (e.g. #objects, space)	Contrast cases	Ranking module Feature Contrast module
Produce and Reflect on an Invention	Generate a solution (e.g. index) and evaluate its correctness	Explain solution	Tell-Me-How module
Math Calculation	Simplify/manipulate fractions	--	Calculator

The two instructional components that help students through the "understand the problem" phase are the "rules tab" and the rule-related hints. Rule-related hints provide feedback exposing inconsistencies in students' inventions. For instance, if a student's invention is not generalizable and only works for specific cases, Oz can provide the following hint: "Don't forget: you have to use the exact same method to find the index for each bus!"

The Invention Coach also supports the "notice and understand features" phase of the Invention process via the "ranking" (described above) and "feature contrast" modules. Ranking the buses from most to least crowded helps students think about why some companies are more crowded than others, which starts to focus them on the features that determine crowdedness. In the "feature contrast" module (Figure 2C), Oz can select two specific buses to contrast. The student is then asked to note which features make one bus more crowded than the other. For example, Oz could ask the student to contrast cases A1 and C2 in Figure 1. Since the number of clowns is held constant across the cases while space changes, the student may begin to notice that clowns alone cannot account for crowdedness, the feature of bus size is important too.

Both “ranking” and “feature contrast” modules employ the instructional strategy of comparing and contrasting cases, to scaffold learners in noticing key features of the problem space.

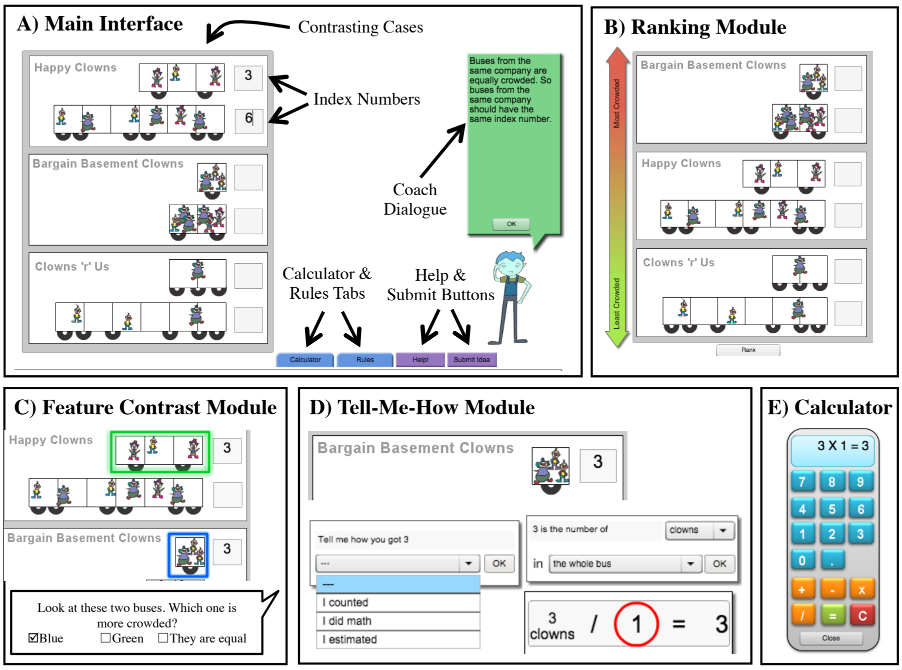

Fig. 2. Prototype Interface and Modules.

The backbone of the Invention Coach is the “tell-me-how” module (Figure 2D), where students are asked to enter and explain their inventions. This serves to recreate the “produce and reflect on an invention” phase of the process while encouraging students to monitor their own errors. In this module, students explain how they arrived at their answer (by selecting whether they “counted,” “estimated,” or “used math”). Students who indicate that they “counted” are further prompted to identify what exactly they counted, while students who “used math” must then use a calculator feature to show how they derived their answers. Students are never provided with direct right/wrong feedback on their solutions. Instead, the tell-me-how module encourages students to explain how they arrived at their solutions, right or wrong. We hope that in the process of explaining their answers, and connecting the math to referents in the cases, students will begin to reflect on their answers and identify gaps in their own understanding. Another key function of this module is to help Oz (and eventually the fully adaptive system) understand how a student generated her index so it can determine appropriate feedback.

Finally, to enable the math calculation phase of the Invention process, students are provided with a calculator (Figure 2E). In our study of naturalistic teacher guidance, many students had difficulty engaging in simple math (e.g. 6 divided by 3), and a large proportion of teacher talk focused on math calculations such as simplifying fractions. The calculator enables students to off-load some of this challenging calculation work and instead focus on the larger concepts behind the math. The "calculator" tab is available in the main interface for students to call up at any time during the task. A calculator is also part of the "tell-me-how" module as described above.

Throughout the phases of the Invention process, the Coach's feedback points out inconsistencies in students' problem solutions. Instead of providing right/wrong or elaborative feedback when students create an incorrect invention, the Coach presents information to contradict the wrong invention. For instance, the Coach may remind the student that their Invention must generalize to all cases or that it must account for two cases that have the same crowdedness. The coach may also present pairs of cases that directly contradict the student. For instance, if a student believes that an irrelevant feature is important, the Coach will show two cases where the irrelevant feature varies but crowdedness does not. This type of feedback enables students to explore on their own, while encouraging them to self-monitor errors and "see" deep features.

In our current design, several components of the Invention Coach must be selected by Oz, while some intelligence is built into the system. The Oz selects whether to respond to a request for feedback by launching a student into a module (e.g. feature contrast, tell-me-how, or ranking) or by giving a single hint, adapting the path through the Invention space based on each student's individual needs. However, once inside a module, the system largely controls the interaction by selecting appropriate feedback and prompting the student to take action.

4 Discussion and Conclusion

We have described the design of a computer-based Invention Coach, which was inspired by a study of naturalistic teacher guidance of paper-based Invention and by prior research on the mechanisms behind Invention. The Invention Coach contains instructional components to address each phase in the Invention process, which can be adaptively selected. The system employs three instructional strategies that target key mechanisms in learning from Invention: contrasting cases, self-explanation of problem solutions, and feedback that exposes inconsistencies in students' solutions. While we are currently implementing a Wizard-of-Oz version of the Invention Coach, we ultimately aim to develop a fully adaptive system.

This work contributes more broadly to work on Invention and exploratory learning environments. To the best of our knowledge, the work presented here is the first design of a guided environment for Invention activities that is based on a model of naturalistic teacher guidance. Our design offers insight into possible strategies and phases of guidance that could be more broadly applicable in other exploratory learning environments and tasks. Specifically, if the Invention Coach we've built proves successful, it would argue that unguided exploration can be augmented by guidance that highlights inconsistencies in student work, contrasts cases to make relevant features salient, and invites students to explain their solutions. These forms of guidance may prove especially useful for developers who wish to retain the emphasis on active processing and construction of ideas inherent in exploratory learning environments, while avoiding the pitfall of unproductive aimless exploration [2, 3].

References

1. Mayer, R. E.: Should There Be a Three-strikes Rule Against Pure Discovery Learning?. American Psychologist, 59(1), 14-19 (2004)
2. Mavrikis, M., Gutierrez-Santos, S., Geraniou, E., & Noss, R. (2013). Design requirements, student perception indicators and validation metrics for intelligent exploratory learning environments. *Personal and ubiquitous computing*, *17*(8), 1605-1620.
3. McLaren, B. M., Timms, M., Weihnacht, D., & Brenner, D. Exploring the Assistance Dilemma in an Inquiry Learning Environment for Evolution Theory. In the Proceedings of the Workshop of Intelligent Support for Exploratory Environments 2012 (ITS 2012).
4. De Jong, T.: Scaffolds for Scientific Discovery Learning. Handling Complexity in Learning Environments: Research and Theory, 107-128 (2006)
5. Kapur, M., & Bielaczyc, K.: Designing for Productive Failure. Journal of the Learning Sciences, 21(1), 45-83 (2012)
6. 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)
7. Bransford, J. D., Franks, J.J., Vye, N.J., Sherwood, R.D.: New Approaches to Instruction: Because Wisdom Can't be Told. Similarity and analogical reasoning, 470 (1989)
8. Schwartz, D. L., Bransford, J. D.: A Time for Telling. Cognition and Instruction 16(4), 475-5223 (1998)
9. Schwartz, D. L., Chase, C. C., Oppezzo, M. A., Chin, D. B.: Practicing Versus Inventing with Contrasting Cases: The Effects of Telling First on Learning and Transfer. Journal of Educational Psychology 103(4), 759 (2011)
10. Shemwell, J. T., Chase, C. C., Schwartz, D. L.: Seeking the General Explanation: A Test of Inductive Activities for Learning and Transfer. Journal of Research in Science Teaching 52(1), 58-83 (2015)
11. Roll, I., Aleven, V., Koedinger, K. R.: The Invention Lab: Using a Hybrid of Model Tracing and Constraint-based Modeling to Offer Intelligent Support in Inquiry Environments. In Intelligent Tutoring Systems, pp. 115-124. Springer, Heidelberg (2010)
12. Chase, C.C., Marks, J., Bernett, D., Bradley, M., & Aleven, V.: Towards the Development of an Invention Coach: A Naturalistic Study of Teacher Guidance for an Exploratory Learning Task. In C. Conati, N. Heffernan, A Mitrovic, & F. Verdejo (Eds.), Proceedings of the 17th International Conference on Artificial Intelligence in Education (2015)
13. Chi, M. T., Feltovich, P. J., & Glaser, R.: Categorization and Representation of Physics Problems by Experts and Novices*. Cognitive science,5(2), 121-152 (1981)
14. Loibl, K., & Rummel, N.: Knowing What You Don't Know Makes Failure Productive. Learning and Instruction, 34, 74-85 (2014)
15. Chi, M. T., Leeuw, N., Chiu, M. H., & LaVancher, C.: Eliciting self-explanations improves understanding. Cognitive science, 18(3), 439-477 (1994).
16. 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(2), 105-154 (2009)

Discovering knowledge models in an open-ended educational game using concept formation

Erik Harpstead, Christopher J MacLellan, Vincent Aleven

Human Computer Interaction Institute, Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh, PA, 15232, USA
{eharpste, cmaclell, aleven}@cs.cmu.edu

Abstract. Developing models of the knowledge and skills being exercised in a task is an important component of the design of any instructional environment. Developing these models is a labor intensive process. When working in exploratory and open-ended environments (EOLEs) the difficulty of building a knowledge model is amplified by the amount of freedom afforded to learners within the environment. In this paper we demonstrate a way of accelerating the model development process by applying a concept formation algorithm called TRESTLE. This approach takes structural representations of problem states and integrates them into a hierarchical categorization, which can be used to assign concept labels to states at different grain sizes. We show that when applied to an open-ended educational game, knowledge models developed from concept labels using this process show a better fit to student data than basic hand-authored models. This work demonstrates that it is possible to use machine learning to automatically acquire a knowledge component model from student data in open-ended tasks.

1 Introduction

When designing intelligent instructional support in educational learning environments it is important to have a model of the skills and knowledge employed during problem solving. A common approach to modeling skills in intelligent tutoring systems (ITSs) is knowledge component (KC) modeling [1]. In the KLI Framework a KC is "an acquired unit of cognitive function or structure that can be inferred from performance on a set of related tasks" [2]. A KC model is a mapping of each problem-solving step in a particular educational environment to the skills necessary to solve that step. KC models can be used in pedagogical software to drive feedback and hints, guide problem selection [3], and inform redesign of the interface [4].

While KC models are useful for a number of purposes in the development of intelligent software they take significant effort to develop. The process of creating a KC model often employs elements of empirical and theoretical task analyses [1], soliciting expert feedback and rationally constructing the skills used in a task. When working in exploratory and open-ended environments (EOLEs) this process is aggravated by the freedom learners experience in these environments. It can be assumed that as the space learners are allowed to explore grows, so too must a KC model grow to continue to provide useful support and feedback to learners. In addition to providing large spaces for exploration, EOLEs often contain more complex representations of domains making it more difficult to articulate the rules defining the applicability of a given KC.

To address the challenges of KC model creation in EOLEs we have developed a novel method for generating new KC models based only on problem states taken from the learning environment. Our approach uses a form of automated model discovery that employs a concept formation algorithm called TRESTLE [5]. This algorithm creates a hierarchical categorization tree based on training examples, which can then be used to label problem states at various grain sizes. The algorithm is designed to handle messy, mixed representations of data, making it ideal for application to EOLEs. It has previously been shown to create clusters similar to humans [5]. In this paper, we show how the conceptual patterns learned by TRESTLE can be used to discover new KC models in the open-ended educational game *RumbleBlocks* [6]. Finally, we conclude with a brief discussion of the implications of this approach and detail how we plan to expand it in future work.

2 The TRESTLE Algorithm

TRESTLE [5] is an incremental concept formation algorithm that creates a hierarchical categorization tree from a set of structured instances. In this section we briefly describe the algorithm's major structures and categorization procedure for more details see [5][1].

The TRESTLE algorithm produces a categorization tree and functions over a set of instances, each described by a set of attribute-value pairs. Instance attributes can have nominal, numeric, or component values that have their own sub-attributes and values. When integrating a new instance TRESTLE proceeds through 3 major steps:

1. Partial Matching, which renames instance attributes to align with the algorithm's current domain understanding
2. Flattening, which converts structured attributes to unstructured ones, while preserving structural information.
3. Categorization, which incorporates the instance into the knowledge base.

TRESTLE's knowledge base is an evolving category structure being built from training examples and is organized into a hierarchical tree of concepts. In building its tree, the algorithm optimizes for a heuristic called category utility, which is similar to maximizing for the expected number of correct guesses that a given concept could make about the attribute-values of a given instance. During categorization new instances are sorted into the tree. At each node in the categorization tree TRESTLE considers 4 different operations and performs whichever one would result in the highest category utility: (1) adding the instance to the best child, (2) creating a new node

[1] A reference implementation is available at: https://github.com/cmaclell/concept_formation

for the instance, (3) merging the best 2 nodes and adding the instance to the result, or (4) splitting the best node by promoting its children to be children of the current node.

After categorizing an instance into its knowledge base, TRESTLE returns a concept label for the instance. Since concepts in TRESTLE are organized in a hierarchical tree, the cluster labels returned from categorization can be generalized if more coarse clusters are desired. At the coarsest, i.e. the root of the tree, everything is considered to be the same concept, while at the most specific, i.e. the leaves of the tree, everything is considered to be unique.

To arrive at a KC label for a step, the problem state in which the step too place is categorized and label is generated based on the returned concept and a desired depth. For a given depth model the state is categorized down the TRESTLE tree. Once the state reaches the desired depth the current concept's label is returned. If the state reaches a leaf of the tree before reaching the desired depth, then the label of the deepest node is used instead. When generating KC models this allows for the creation of multiple model variants that consider the domain at different levels of granularity (see **Fig. 1**).

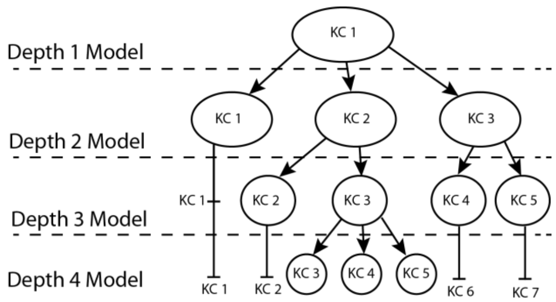

Fig. 1. A diagram of how KC labels are attributed to problem states based on their categorization in the TRESTLE tree for a given depth mode.

3 *RumbleBlocks*

To demonstrate how TRESTLE can be used to aid in the process of KC modeling we introduce *RumbleBlocks* [6], an open-ended educational game. *RumbleBlocks* is a physics game designed to teach children (ages 5-8) three basic concepts of structural stability and balance: (1) objects with wider bases are more stable, (2) objects that are symmetrical are more stable, and (3) objects with lower centers of mass are more stable.

In the game, players are tasked with building a tower out of blocks to help a stranded alien power their spaceship (see **Fig. 2**). The tower must be tall enough to reach the alien and cover a series of energy orbs that power the spaceship. Once players have finished building their tower they place the spaceship on top, which triggers an earthquake. If, after the earthquake, the ship is still on top of the tower, then the player has succeeded and advances on to the next level, otherwise they must try the level again.

Fig. 2. A screenshot of *RumbleBlocks*

Each level in *RumbleBlocks* is designed to emphasize one of the three key concepts of stability. This emphasis is accomplished through the placement of energy orbs, the target zone for the spaceship, and the palette of available blocks. While each level is targeted at a particular principle, there is a wide range of variance in the kinds of solutions players design to in-game challenges. Our previous analysis found that there are several levels where less than 10% of students actually used the solution envisioned by the game's designers [7]. The variance in player behavior demonstrates the open-endedness of the game as well as highlights the challenge inherent in defining KC models to measure learning in the game.

4 KC Model Discovery in *RumbleBlocks*

To evaluate the application of TRESTLE to the KC modeling process we used it to discover a set of new KC models in *RumbleBlocks*. For comparison we also created a "hand built" KC model meant to capture the original design intent behind the game. This model labels each level in the game with the principle it is designed to emphasize. Since the first 5 levels of the game are primarily a mechanical tutorial for the game rather than instructional levels dealing with physics principles, we relabeled these levels with an "Intro" KC, resulting in a hand-built model with 4 KCs.

For this first demonstration of the use of TRESTLE to generate KC models we chose to focus on a broad definition of a step as solving an entire level of *RumbleBlocks*. This is in keeping when Van Lehn *et al.*'s definition of a step as "the smallest possible correct entry that a student can make" [8] because, in its current form, *RumbleBlocks* only provides correctness feedback to players at the end of a level. In this context a step is then considered in terms of the initial level state given to the player to construct a solution in and evaluated based on their final construction. The state representation used for training TRESTLE contained the positions of each of the energy orbs, the target position of the spaceship, and the available number of each block type. The resulting categorization tree, based on the initial state data from *Rumble-*

Blocks' 47 levels, was 7 levels deep giving us 7 candidate KC models each with differing levels of granularity.

To evaluate relative appropriateness of different candidate KC models we used the tool suite provided by DataShop [9]. In particular, we used AFM [10], a specialized form of logistic regression that fits a given KC model to student log data. The resulting regression model can be used to assess the fit of a particular KC to the real student data. DataShop provides several model fit statistics to compare KC models: AIC and BIC, both standard model fit statistics that penalize for model complexity and Cross Validated Root Mean Square Error (CV-RMSE) using 3-fold cross validation with different stratification schemes (i.e. student, item and un-stratified).

The data we use in our evaluation comes from a formative evaluation of the game with 174 players in the target demographic. Players were allowed to play the game for two 20-minute sessions.

The model fit estimates for the 7 Trestle-based models and the original Principle (i.e., hand-built) model can be seen in **Table 1**. In general, more fine grained models tend to fit the data better. The TRES-Depth7 model is preferred according to AIC and both item-stratified and un-stratified RMSE. This would suggest that an appropriate model for initial states in *RumbleBlocks* is one that treats all levels as nearly unique from each other.

Table 1. Fit statistics for each KC model. Cross Validated Root Mean Square Errors (CV-RMSE) are based on 3 fold cross validation using different forms of stratification.

Model	*KCs*	*AIC*	*BIC*	*CV-RMSE (student)*	*CV-RMSE (item)*	*CV-RMSE (none)*
Principle	4	6560.73	**8544.74**	.3856	.3883	.3869
TRES-Depth1	1	6828.35	8771.45	.3924	.3948	*NA*
TRES-Depth2	5	6737.21	8734.85	.3899	.3921	.3923
TRES-Depth3	14	6661.67	8782.03	.3878	.3904	.3915
TRES-Depth4	24	6530.78	8787.50	.3845	.3853	.3855
TRES-Depth5	32	6350.50	8716.31	.3794	.3821	.3826
TRES-Depth6	39	6152.75	8614.01	**.3734**	.3761	.3739
TRES-Depth7	41	**6152.28**	8640.81	.3736	**.3754**	**.3732**

5 Discussion

We can see from the results that KC models based on depth cuts of a TRESTLE categorization tree better fit student data than a model based on the original design of the game in terms of AIC and cross-validation. According to these statistics, we find that a more specific KC model better fits student data than more general models. This would make it appear that there is little transfer going on within the game. However, this is likely due to our unit of analysis. An approach that employs a more fine grained definition of a correct step (e.g., steps defined at the transaction level) might reach a different conclusion with regards to transfer because there is likely to be some common application of knowledge components used across building towers in different levels.

The approach presented here deals with concept granularity at a holistic level. By this we mean that all KCs in a model are being considered at the same depth of the concept tree. There is some evidence that suggests human learners will employ concepts at different levels of granularity based on their expertise [11]. It is possible that the most appropriate KC model uses a combination of specific and general concepts depending on the context of the task at hand. Rather than creating KC labels as uniform cuts of a concept hierarchy, where concepts all exist at the same depth, we could instead start all problem states at their coarsest label and iteratively split concept nodes into more specific labels. After each split the resulting KC model could be tested for fit using student data until an optimal model is found. This is similar to the Learning Factors Analysis search algorithm [12] but it would not require human developed models as seeds. Exploring this process is something we look forward to in future work.

Our current analysis defined steps to be the complete solution to each level. This follows with Van Lehn *et al.*'s definition of a step in KC analysis as the smallest amount of action that a student can perform correctly [8]. This definition still assumes that all possible solutions to a level exercise the same skill, which may not be the case in practice. One way of going beyond this assumption in analyzing *RumbleBlocks* is to create a TRESTLE model based on the solutions players make to each in-game level rather than the initial conditions of the level. Such an approach would allow for analysis according to different kinds of solutions rather than the constraints under which problem solving took place. One issue with taking into account the content of students' solutions is how to handle the assignment of KC labels when there are multiple valid solutions to a level, as is the case with *RumbleBlocks* [7]. In the case of correct solutions it is simple to state that each unique correct solution embodies the use of a different KC. When looking at incorrect solutions, however, the question of attribution becomes more difficult as it is hard to know which of the possible correct approaches the student failed to execute correctly. A standard modeling approach would assign an incorrect step with the labels of all possible correct solutions; using a variant to AFM's statistical formula to allow for the disjunction of KCs [10]. A TRESTLE based approach could go beyond this by categorizing incorrect solutions into a knowledge base trained on correct solutions and assigning a KC label based on which correct solution the error most closely resembles. This is similar to the approach taken by Rivers and Koedinger to create next step feedback in programing tasks [3] but has the potential to be domain general. Exploring this approach to KC modeling with TRESTLE remains a topic of our future work.

Ideally, we would like to go beyond the final state definition of a step to a transaction-level model. Having a full transaction-level model would allow for the inclusion of targeted feedback to players while they are playing rather than providing feedback only at the end of building. Additionally, more detailed understanding of player problem solving could better inform adaptive sequencing. The challenge in taking this approach in *RumbleBlocks* is that that evaluation of player performance is currently only performed at the end of a level. This creates similar correctness attribution challenges in deciding whether a particular build step is a good or bad example of a given concept. Again we could turn to TRESTLE to aid in this analysis by having it perform categorization on whole solution paths rather than final solutions. There are several open questions with this analysis in terms of how best to represent a solution path for categorization but we hope to resolve these issues in future work.

6 Conclusion

This paper presents a preliminary use of TRESTLE as a way to discover new KC models in an open-ended game. The models automatically discovered by TRESTLE better fit student data than one hand-built to capture the design intent of the game. This demonstrates the promise of concept formation based approaches to KC model creation. In future work we plan to further explore the implications of TRESTLE-based KC models including discovering transaction-level models and exploring models that capture mixed grain sizes. We hope other researchers can find utility in these methods and apply them to their own exploratory and open-ended environments.

7 Acknowledgement

We would like to thank the developers of *RumbleBlocks* and our colleagues who performed the evaluation that provided our data. This work was supported in part by the DARPA ENGAGE research program under ONR Contract Number N00014-12-C-0284 and by a Graduate Training Grant awarded to Carnegie Mellon University by the Department of Education # R305B090023.

8 References

1. Aleven, V., Koedinger, K.R.: Knowledge Component Approaches to Learner Modeling. In: Sottilare, R.A., Graesser, A., Hu, X., and Holden, H. (eds.) Design Recommendations for Intelligent Tutoring Systems: Volume 1 - Learner Modeling. pp. 165–182. U.S. Army Resarch Laboratory (2013).
2. Koedinger, K.R., Corbett, A.T., Perfetti, C.: The Knowledge-Learning-Instruction Framework: Bridging the Science-Practice Chasm to Enhance Robust Student Learning. Cogn. Sci. 36, 757–98 (2012).
3. Rivers, K., Koedinger, K.R.: Automating Hint Generation with Solution Space Path Construction. Proc. ITS 2014. pp. 329–339 (2014).
4. Koedinger, K.R., Stamper, J.C., Mclaughlin, E.A., Nixon, T.: Using Data-Driven Discovery of Better Student Models to Improve Student Learning. Proc. AIED 2013. pp. 421–430 (2013).
5. Maclellan, C.J., Harpstead, E., Aleven, V., Koedinger, K.R.: TRESTLE: Incremental Learning in Structured Domains using Partial Matching and Categorization. Proceedings of the 3rd Annual Conference on Advances in Cognitive Systems - ACS 2015 (2015).
6. Christel, M.G., Stevens, S.M., Maher, B.S., Brice, S., Champer, M., Jayapalan, L., Chen, Q., Jin, J., Hausmann, D., Bastida, N., Zhang, X., Aleven, V., Koedinger, K., Chase, C.,

Harpstead, E., Lomas, D.: RumbleBlocks: Teaching science concepts to young children through a unity game. Proc. CGAMES 2012. pp. 162–166. IEEE (2012).

7. Harpstead, E., Maclellan, C.J., Koedinger, K.R., Aleven, V., Dow, S.P., Myers, B.A.: Investigating the Solution Space of an Open-Ended Educational Game Using Conceptual Feature Extraction. Proc. EDM 2013. pp. 51–58 (2013).
8. Vanlehn, K., Koedinger, K.R., Skogsholm, A., Nwaigwe, A., Hausmann, R.G.M., Weinstein, A., Billings, B.: What's in a Step? Toward General, Abstract Representations of Tutoring System Log Data. Proceedings of the 11th International Conference on User Modeling - UM 2007. pp. 455–459. Springer Berlin Heidelberg (2007).
9. 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., and Baker, R.S.J. d. (eds.) Handbook of Educational Data Mining. pp. 43–55 (2010).
10. Cen, H., Koedinger, K., Junker, B.: Comparing Two IRT models for conjunctive skills. Proc. ITS 2008. pp. 796–798 (2008).
11. Fisher, D.: A Computational Account of Basic Level and Typicality Effects. Proc. 1988 Nat. Conf. Artificial Intelligence (AAAI'88). pp. 233–238 (1988).
12. Cen, H., Koedinger, K., Junker, B.: Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement. Proceedings of the 8th International Confernence on Intelligent Tutoring Systems - ITS 2006. pp. 164–175 (2006).

Towards Configurable Learning Analytics for Constructionist Mathematical e-Books

Sokratis Karkalas, Christian Bokhove, Patricia Charlton, and Manolis Mavrikis

London Knowledge Lab, Institute of Education, University College London, UK
s.karkalas,c.bokhove,m.mavrikis,p.charlton@ioe.ac.uk

Abstract. This paper presents emerging requirements for learning analytics on interactive mathematical e-books and a framework that can be used for the seamless integration of complex learning objects with e-book platforms. We describe the opportunities that this approach opens up regarding interoperability and configurability of learning analytics and intelligent support. The framework is generic and can be used for any type of system with similar requirements. In this paper we present a case that covers configuration of learning analytics for teachers and intelligent support for students in constructionist mathematical e-books.

1 Introduction

The emergence of authoring software for e-books means that digital books with text, images and other interactive elements are increasingly being used on personal computers and other electronic devices for educational purposes. However, most of these e-books are simple transformations of traditional textbooks into a digital format and do not take advantage of the dynamic and computational affordances offered by this emerging technology. The MCSquared project[1] is investigating whether the affordances of state-of-the-art e-books can be exploited to support the learning of abstract mathematical concepts. We are looking into the design of highly interactive constructionist e-book widgets, and exploring their potential for providing learners with opportunities to construct mathematical artefacts in order to engage creatively with mathematical problems.

Within this context the increase of both process and product data collected provides unprecedented opportunities for knowledge discovery through state-of-the-art data analysis and visualization techniques. However, despite the fact that in the past two decades intelligent technology has become increasingly feasible, the power of these methods has not reached its full potential in education. For example, although it is now possible for intelligent pedagogical agents to monitor learners' interactions within educational applications and provide individualised support, only a handful of intelligent tools are employed in practice, yet they are tied to particular instructional approaches, domains and context.

[1] The Mathematical Creativity Squared project is funded by the EU, under FP7 ICT-2013.8.1 Project #610467. For more details see http://www.mc2-project.eu

We believe that one of the reasons that the promises of ubiquitous, individualised and adaptive technology has had a very small impact in education is that learning environments are often rigid and limited to specific learning contexts and pedagogical approaches. Our previous research [6, 7] and that of others (e.g., [9]) has primarily enabled the rapid revision and management of content. In line with previous research in the field (e.g. [8]), our vision is that teachers and educational organizations will be able to also mould the nature and type of support provided to a learner (cf. [2, 10]) and the information they want to glean from their interaction. Then the unrealised potential of the technology could begin to be exploited.

This paper presents our preliminary efforts towards this vision: a prototype where e-book pages and the widgets that they contain can be configured. First, we present below a set of emerging requirements for Learning Analytics in the context of constructionist mathematical e-books.

2 Emerging Requirements for Configurable Learning Analytics

With the advent of data science and analytics in general, there are several 'analytics' tools that have appeared. While we have looked into a large subset of them, we cannot review them all in detail here. However, we have been unable to find a tool that focuses on providing information from constructionist, exploratory mathematical environments (with the exception of our previous work in [3] where we also review related work in more detail).

In the context of commercial e-books in particular publishers and authors are interested in (and to some extent only have access to) high level information such the number of pages read, average reading times, exit rates and other details that reveal reading patterns that can correlate with, for example, sales figures. However, from an educational point of view teachers, designers and even students require a more in-depth analysis of learners' interaction with the e-books.

The MCSquared project comprises four Communities of Interest (COI) across 4 EU countries (France, Greece, Spain, UK) and engaged their members in requirements elicitation and stakeholders' analysis. Through several face-to-face workshops and sustained online interaction and communication between members of the COI we have identified many scenarios in which e-books are being used in teaching and other requirements of learning analytics tools and data visualisation that are emerging.

Digital resources like e-books are being used either directly in the classroom or in 'blended' learning scenarios (e.g. for practice exercises at home) or in a 'flipped' learning model where students read and interact with the e-book content online (e.g. at home) and complete other parts of the e-book in the classroom with the help of other students or the teacher. So neither context can be excluded. We present below high-level categories of the themes around which requirements have emerged:

- Usage and other book-level descriptive statistics

  - the order of pages
  - time spent on each page/activity
  - how quickly students read a page
  - the percentage of coverage of particular pages from a book
- Structured answer and related descriptive statistics
  - Student answers and performance in structured questions
  - Number of attempts to answer a question
  - Repeated wrong answers across students
- Constructionist Analytics
  - Constructionist descriptive statistics (i.e. number of objects constructed, moved, deleted, etc.)
  - Data regarding construction operations (achievements of key 'landmarks')
  - Specific patterns of interaction within a widget

While the first and second category of data analytics are interesting in their own right, we are focusing mostly on the third type of data that we refer to as 'deep' analytics of constructionist e-books for learning. This is particularly interesting because it goes beyond the 'low-hanging fruit' of descriptive statistics (which, in principle, are technically and conceptually well understood) and looks into extracting some meaningful information that could support decision making. Constructionist analytics opens up the door to real-time formative and summative assessment (as discussed in [1]). In our previous work, we found that even a simple traffic-light system could satisfy the teacher's need for finding out which students are progressing satisfactorily towards completing the task and which ones may be in difficulty [3].

In addition, a requirement across all the types of analysis mentioned above, is the availability of a generic, interoperable framework that enables configurability of learning analytics and intelligent support. We present a prototype of this in the next section.

3 Prototype

In this section, using an example of an e-book page, we demonstrate a basic but complete integration scenario. The page is part of a mathematics e-book developed by the Greek COI and features a learning activity developed in Geogebra. The page is integrated in a prototype that has a local in-memory database that stores data generated from the student activities and a rule-based reasoner that provides real-time intelligent support to the students (fig. 1). The purpose of the activity is to get the student select an appropriate combination of variables in order to get both parts of the ladder to the same level. Converging the two parts can then display a single heart at the top (join the two halves). All of these heterogeneous components are pluggable widgets that operate in their own secure environment (sandbox). They are hosted in their own domains and they are executed concurrently without interfering with one another. Integration with the host page takes place through a lightweight set of mediator

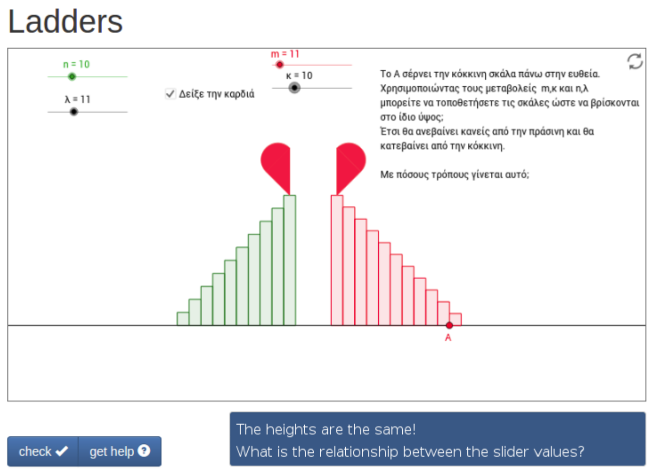

Fig. 1. The 'Ladders' Activity from Geogebra Tube

wrappers that enable full two-way communication over a simple and common interface. Each widget is allowed to expose its own functionality (or part of it) and make it available to the platform through a wrapper interface. This scheme allows better performance (multithreading), security (sandboxing), controllable interoperability (widget interface exposed through the wrapper) and seamless integration (common wrapper interface) [5].

This e-book page demonstrates an example of an activity that offers real-time intelligent support to students through visual controls and real-time formative and summative feedback to teachers through graphs. The activity widget offers interactivity through sliders and a checkbox. As the student interacts with the widget, action indicators are generated and sent to the page. The platform populates the local (in-memory) database which in turn incrementally synchronises with the back-end database through REST [2] web-service endpoints (fig. 2). These updates are asynchronous for better performance. The local database serves as a buffer for data that needs to be immediately available and thus enables fast and more reliable responses. The local data is then sent to the rule-based reasoner for processing. If the reasoner identifies a case that justifies a discreet intervention, a message is displayed in the textbox and/or some visual indicator is presented in the activity frame (heart). The latter presupposes that messages are sent to the activity widget through the platform. This process may also be initiated by the student. If the student asks for help or wants the system to evaluate the work that has been submitted so far, then the reasoner responds with an appropriate message in the textbox. In parallel, the data generated from both the activity and the reasoner is sent to the database.

[2] http://en.wikipedia.org/wiki/Representational_state_transfer

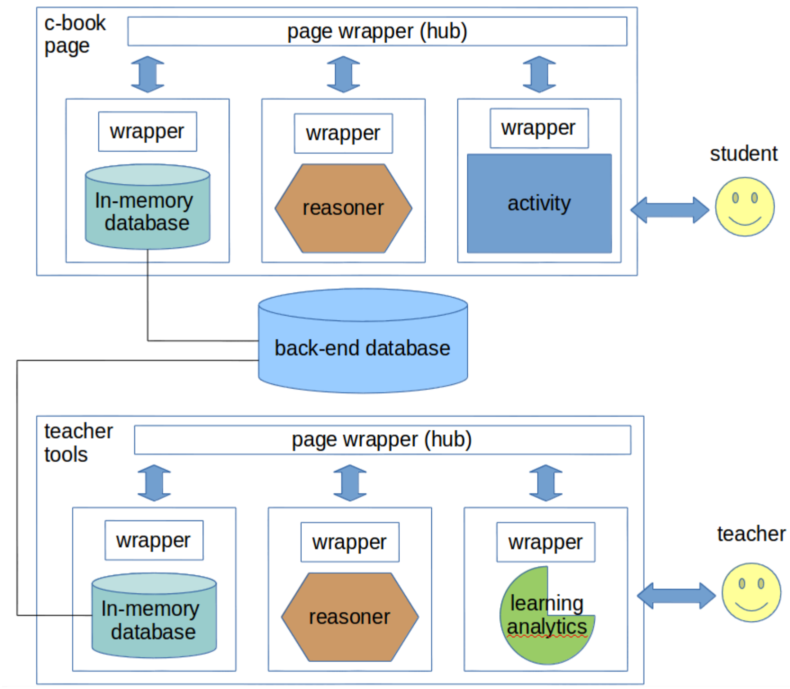

Fig. 2. The Architecture

The page that hosts the teacher tools has a similar structure. It also contains a local database widget and a reasoner. As the back-end database gets updated with student actions and reasoner findings, the local database incrementally retrieves the changes. Some of this data is used as a direct feed to other widgets that host learning analytics visualisations. In this particular example we have visualisations that measure student activity and performance (fig. 4). Both measurements are presented as histograms and provide real-time feedback to the teacher. The first visualisation measures what has been used in the activity and how much. For example the elements n, m, k and are numeric variables that correspond to sliders in the construction. The visualisation shows which of these sliders and how many times have been used by the student. The second visualisation presents a comparative measurement of effort and levels of achievement. Some of the local data is then processed by the reasoner and new data may be inserted into the database. This data may be used to populate other visualisations or provide some intelligent support to the teacher.

The teacher tool is both an authoring and a monitoring environment. The teacher has the ability to dynamically configure the system to log actions performed by specific widget elements. Widget instances can be dynamically inserted into the authoring environment in the same way they can be integrated with a c-book page. The widget communicates with its host through the wrappers and makes available its internal structure to it. The metadata extracted from the widget is then used by the host to dynamically construct an authoring graphical user interface that is presented to the teacher (fig. 3). The teacher can then select the widget elements deemed necessary to log their actions. This information is sent to the database along with the id of the c-book the widget belongs to. When the widget is invoked in a c-book, the page uses this information to dynamically register event handlers in the widget in order to intercept student actions for the selected elements.

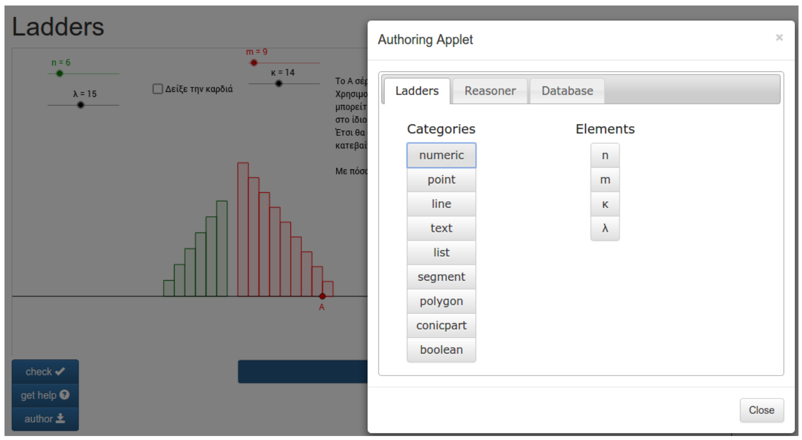

Fig. 3. Authoring Applet for the Teacher

4 Conclusion and Future Work

In this paper we presented a prototype authoring environment that enables configurable learning analytics and intelligent support in educational e-books. The specific example used in this presentation focuses on constructionist mathematical learning activities and the configuration of appropriate analytics for them. The system has been implemented and used by members of COIs and preliminary results show that it meets its original design objectives. It can be used effectively for rapid integration of learning objects and dynamic configuration of learning analytics and intelligent support. The next step is to specify how this data will be processed by the reasoner in order to provide effective support to the students. This part requires the use of a rule editor by a domain expert. Preliminary work towards this aspect has been undertaken in [4].

A distinguishing characteristic of the prototype presented here is the ability to dynamically generate user interfaces that enable the configuration of learning analytics on heterogeneous learning objects. Heterogeneity is hidden behind the mediator wrappers. A possible future enhancement would be to analyse a number

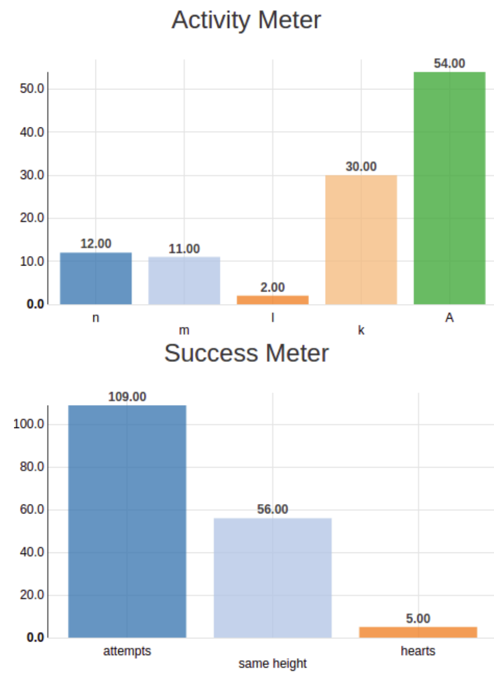

Fig. 4. Teacher Visualisations. In this example the x axis represents the different aspects that the teacher selected to log and the y-axis the number of logged cases.

of representative learning objects and create a learning component description language that can be used as a standard description of the construction that represents an activity. This language could then be used to semantically enhance the component in the wrapper in a standardised way.

References

1. Berland, M., Baker, R., Blikstein, P.: Educational data mining and learning analytics: Applications to constructionist research. Technology, Knowledge, and Learning (19), 205–220 (2014)
2. Bokhove, C.: Implementing feedback in a digital tool for symbol sense. International Journal for Technology in Mathematics Education 17(3), 121–126 (2010)
3. Gutierrez-Santos, S., Mavrikis, M., Geraniou, E., Poulovassilis, A.: Usage scenarios and evaluation of teacher assistance tools for exploratory learning environments. Computers and Education (in press)
4. Karkalas, S., Gutierrez-Santos, S.: Enhanced javascript learning using code quality tools and a rule-based system in the flip exploratory learning environment. In: Advanced Learning Technologies (ICALT), 2014 IEEE 14th International Conference on. pp. 84–88. IEEE (2014)
5. Karkalas, S., Mavrikis, M., Charlton, P.: Turning web content into learning content. a lightweight integration and interoperability technique (2015), under review

6. Mavrikis, M., González Palomo, A.: Mathematical, interactive exercise generation from static documents. Electronic Notes in Theoretical Computer Science 93, 183–201 (Feb 2004), http://www.lkl.ac.uk/manolis/pubs/pdfs/mkm-interactive-04.pdf
7. Mavrikis, M., Maciocia, A.: Wallis: a web-based ILE for science and engineering students studying mathematics. three years on. (2006), http://webalt.math.helsinki.fi/webalt2006
8. Murray, T., Blessing, S., Ainsworth, S.: Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software. Springer (2003)
9. Pardo, A., Fisteus, J.A., Kloos, C.D.: A distributed collaborative system for flexible learning content production and management. Journal of Research and Practice in Information Technology 44(2), 203 (2012)
10. Sangwin, C.: Automating the marking of core calculus and algebra: eight years on. In: Robinson, M., Challis, N., , Thomlinson, M. (eds.) Maths at University, pp. 135–139. More Maths Grads (2010)

Analysing Project Based Learning Scenarios to inform the design of Learning Analytics: Learning from related concepts

Rose Luckin, Manolis Mavrikis, Katerina Avramides, Mutlu Cukurova

London Knowledge Lab, UCL Institute of Education, University College London

Abstract. Project Based Learning is a complex concept that is related to Problem Based Learning and Collaborative Problem Solving. These latter concepts are well represented in the literature by models and frameworks that can usefully be adapted to develop a framework for the analysis of Project Based Learning. We present such a framework that has been designed for learning situations that involve the use of technology. This technology can be used to capture data about learners' interactions as well as to support their learning. We suggest that this data can be combined with data collated by human observers and analysed using the framework.

Introduction

The literature on Project Based Learning is complex with many related concepts, for example: Practice Based Learning, Problem Based Learning, Collaborative Problem Solving and Inquiry Learning. In this paper we explore the frameworks for two of these concepts: Problem Based Learning (PBL) and Collaborative Problem Solving (CPS) in an attempt to identify a framework for the analysis of Project Based Learning activities to inform the design of Learning Analytics. We have selected these two concepts, because they are well supported by existing models and frameworks.

1.1 Problem Based Learning

Problem based approaches encourage learners to become actively engaged in meaningful real-world problems that often require practical as well as intellectual activity. The premise is that the students who participate in a PBL approach will learn through solving problems together and then reflecting upon their experience (Barrows and Tamblyn, 1980). Problem-based approaches to learning (PBL) are not new, they date back to the early 20th century in the work of Dewey (1938) for example (Hmelo-Silver, 2004). Whilst they were initially part of medical education and law schools; they have recently gained more popularity with educators in schools and universities for teaching STEM subjects. A key element of PBL is that the students work collaboratively, learning from each other and solving the problem together. The teacher's role is that of facilitator, but the students are very much self-directed. The PBL approach therefore requires that participating students have good collaborative skills and sufficient metacognitive awareness to steer them through the problem space in a manner that enables their learning. As a result the potential outcomes for the students are not merely cognitive in terms of their increased understanding of the subject matter of the problem, but also there are advances in the transferable twenty first century skills of communication, collaboration and critical thinking.

Hmelo-Silver (2004) uses a stepwise model to describe the PBL process from the teacher's perspective (see Figure 1). Students start by identifying relevant facts about the problem, which increases their understanding and enables them to generate their hypotheses about potential solutions. The teacher or potentially a more able peer helps the student to recognize what are referred to as knowledge deficiencies that will become the goals of their self-directed study. Once these knowledge deficiencies have been addressed the student can re-evaluate their hypotheses and learn through a process of reflection and application.

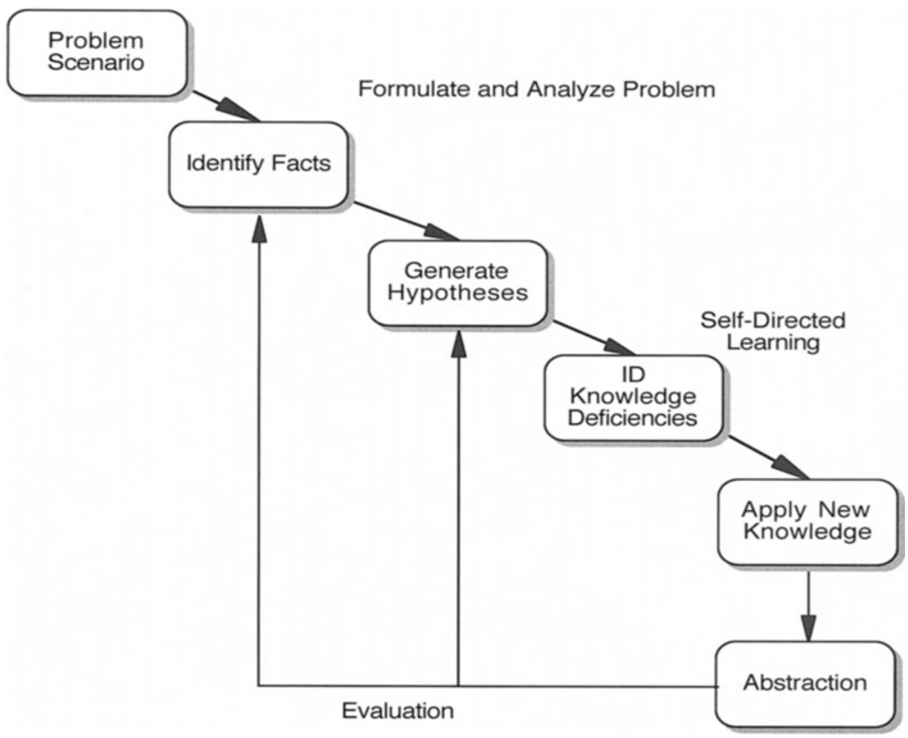

Fig. 1. PBL Tutorial Model (Hmelo-Silver, 2004)

1.2 Collaborative Problem Solving

More recently, and in preparation for the 2015 PISA assessments, the OECD has developed a framework for the assessment of collaborative problem solving (CPS) that is complementary to the traditional PBL approach outlined above (OECD, 2013). The OECD defines CPS as:

> Collaborative problem solving competency is the capacity of an individual to effectively engage in a process whereby two or more agents attempt to solve a problem by sharing the understanding and effort required to come to a solution and pooling their knowledge, skills and efforts to reach that solution.
>
> (OECD, 2013, p.6)

There are three core competencies that are fundamental to this definition of CPS:

1. Establishing and maintaining shared understanding;
2. Taking appropriate action to solve the problem;
3. Establishing and maintaining team organisation.

These are combined with a set of problem solving competencies that are similar to those outlined by Hmelo-Silver (2004), although there is no explicit reference to knowledge deficiencies. This is not surprising because the PBL model is one of tuition, whereas the OECD CPS model is one of assessment:

1. Exploring and Understanding
2. Representing and Formulating
3. Planning and Executing
4. Monitoring and Reflecting

The OECD framework for CPS also includes three further elements:

1. Three conceptual dimensions for the assessment of problem solving. These are the problem context, the nature of the problem situation, and the problem solving process;
2. Two aspects of the problem solving context: the setting (whether or not it is based on technology) and the focus (whether it is personal or social);
3. Two problem presentation types: static problem situations in which the information about the problem situation is complete, and interactive problem situations, where it is necessary for the problem solver to explore the problem situation in order to obtain additional information.

These additional elements highlight the complexity of CPS activities and are pulled together in Figure 2 below.

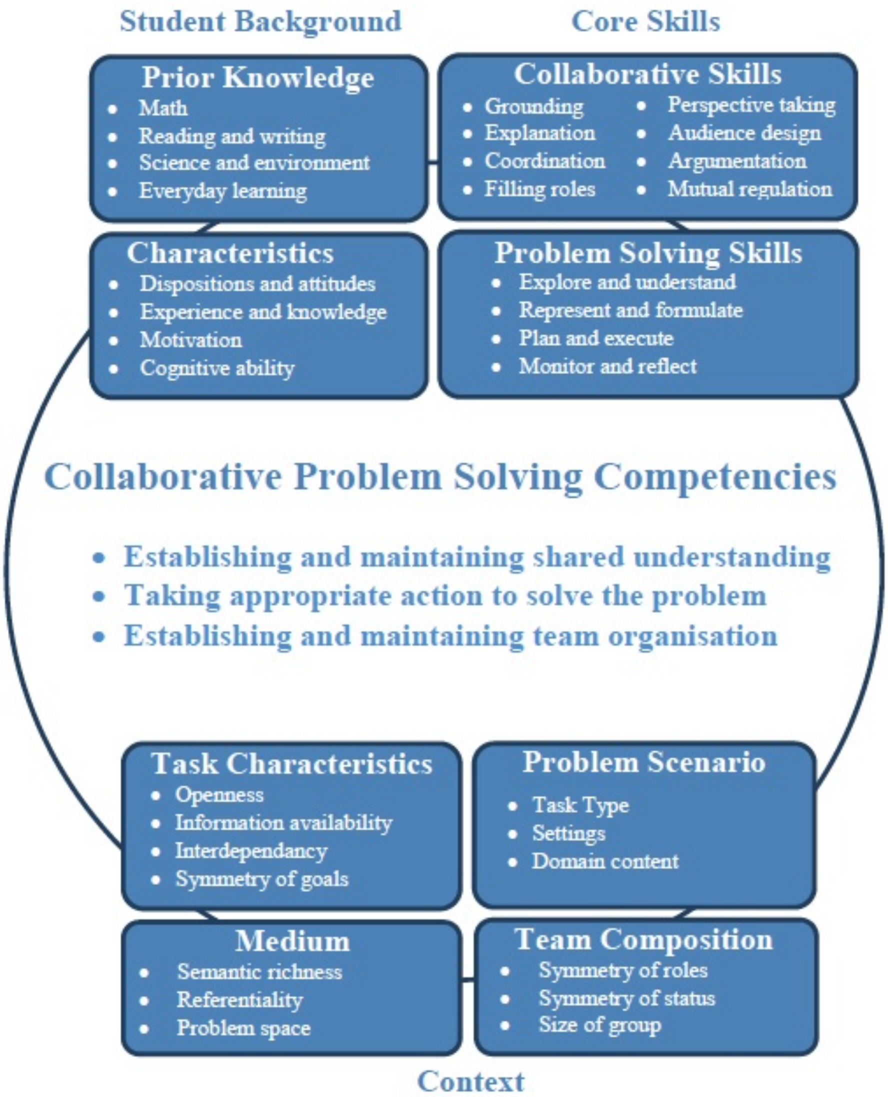

Fig. 2. Overview of factors and processes for Collaborative Problem Solving in PISA 2015

In addition to this overview the four problem solving processes and the three major collaborative problem solving competencies are merged to form a matrix of specific skills, see Table 1. In the resulting matrix, the skills have associated actions, processes, and strategies. These specify what it means for the student to be competent.

	(1) Establishing and maintaining shared understanding	(2) Taking appropriate action to solve the problem	(3) Establishing and maintaining team organisation
(A) Exploring and Understanding	(A1) Discovering perspectives and abilities of team members	(A2) Discovering the type of collaborative interaction to solve the problem, along with goals	(A3) Understanding roles to solve problem
(B) Representing and Formulating	(B1) Building a shared representation and negotiating the meaning of the problem (common ground)	(B2) Identifying and describing tasks to be completed	(B3) Describe roles and team organisation (communication protocol/rules of engagement)
(C) Planning and Executing	(C1) Communicating with team members about the actions to be/ being performed	(C2) Enacting plans	(C3) Following rules of engagement, (e.g., prompting other team members to perform their tasks.)
(D) Monitoring and Reflecting	(D1) Monitoring and repairing the shared understanding	(D2) Monitoring results of actions and evaluating success in solving the problem	D3) Monitoring, providing feedback and adapting the team organisation and roles

Table 1. Matrix of Collaborative Problem Solving skills for PISA 2015

Learning from Problem Based and Collaborative Problem Solving

The type of matrix in Fig. 1 has the potential for use when analyzing data of collaborative activity, but for a PBL approach, the missing component of knowledge deficiency requires attention. In Table 2, we add the PBL tutorial stages to the matrix to address this limitation. In this way we combine a tuition model with an evaluation model and in so doing address both aspects of the teaching learning process.

	(1) Establishing and maintaining shared understanding	**(2) Taking appropriate action to solve the problem**	**(3) Establishing and maintaining team organisation**
(A) Identifying facts	(A1) Discovering perspectives and abilities of team members, making knowledge explicit	(A2) Discovering the type of collaborative interaction to solve the problem, along with goals	(A3) Understanding roles to solve problem
(B) Representing and Formulating	(B1) Building a shared representation and negotiating the meaning of the problem (common ground)	(B2) Identifying and describing tasks to be completed	(B3) Describe roles and team organisation (communication protocol/rules of engagement)
(C) Generating Hypotheses	(C1) Critically analysing the problem representation	(C2) Generating and Communicating potential solution paths	(C3) Present Hypothesis, encourage feedback from others and offer feedback on others' hypotheses
(D) Planning and Executing	(D1) Communicating with team members about the actions to be/ being performed	(D2) Enacting plans	(D3) Following rules of engagement, (e.g., prompting other team members to perform their tasks.)
(E) Identifying Knowledge and Skill Deficiencies	(E1) Comparing the team's knowledge and skills with the proposed actions	(E2) Identifying and specifying individual deficiencies	(E3) Identifying and specifying team deficiencies
(F) Monitoring, Reflecting and Applying	(F1) Monitoring and repairing the shared understanding	(F2) Monitoring results of actions and evaluating success in solving the problem	(F3) Monitoring, providing feedback and adapting the team organisation and roles

Table 2. Combined Matrix that merges PBL and CPS concepts adapted from PISA 2015

Each of the 18 cells can be associated with different levels of learner proficiency. For example;

Low — the student responds to or generates information that has little relevance to the task.

Medium — the student responds to most requests for information and prompts for action, and generally selects actions that contribute to achieving group goals.

High — the student responds to requests for information and prompts for action, and selects actions that contribute to achieving group goals (OECD, 2013).

The contents of the cells C1 to C3 and E1 to E3 have been generated by the authors informed by Hmelo-Silver (2004).

Final Remarks and Further Research

Frameworks such as this offer a flexible approach to the analysis of data collected from project based learning scenarios. This analysis may be that completed by humans as we strive to understand whether and how learning happens, but could it also be useful for data collected and analysed by machine? It needs to be acknowledged that PBL activity may not be captured completely through technology and that there will be aspects of the activity that take place away from any current technology. It may therefore be necessary for any analytics to use a combination of human and machine generated data. Our next steps are to test the framework empirically with a project based data set and to consider what appropriate learning analytic requirements might be extracted. At the workshop we will bring some examples of data and associated analysis to support further discussion of the framework.

Acknowledgements

This work is co-funded by the European Union under the Practice-based Experiential Learning Analytics Research And Support (PELARS) STREP Project of the FP7-ICT-2013-11 Objective ICT-2013.8.2 Technology-enhanced Learnin #619738. See http://www.pelars-project.eu/

References

1. Barrows, H. S., and Tamblyn, R. (1980). Problem-Based Learning: An Approach to Medical Education, Springer, New York.
2. Dewey, J. (1938). Experience and Education, Macmillan, New York.
3. Hmelo-Silver, C. E. (2004) Problem-Based Learning: What and How Do Students Learn? Educational Psychology Review, Vol. 16, No. 3.
4. OECD (2013) Draft Collaborative Problem Solving Framework http://www.oecd.org/pisa/pisaproducts/Draft%20PISA%202015%20Collaborative%20Problem%20Solving%20Framework%20.pdf

Robust student knowledge: Adapting to individual student needs as they explore the concepts and practice the procedures of fractions

Claudia Mazziotti[1], Wayne Holmes[2], Michael Wiedmann[1], Katharina Loibl[3], Nikol Rummel[1], Manolis Mavrikis[2], Alice Hansen[2], Beate Grawemeyer[4]

[1] Ruhr-University Bochum, Germany
{claudia.mazziotti,michael.wiedmann,nikol.rummel}@rub.de
[2] UCL Institute of Education, University College London, UK
{w.holmes, m.marvrikis, a.hansen}@lkl.ac.uk
[3] University of Education Freiburg, Germany
loibl@ph-freiburg.de
[4] Birkbeck College, University of London, UK
beate@dcs.bbk.ac.uk

Abstract.
Robust knowledge consists of both conceptual *and* procedural knowledge. In order to address both types of knowledge, offering students opportunities to explore target concepts in an exploratory learning environment (ELE) is insufficient. Instead, we need to combine exploratory learning environments, to support students acquisition of conceptual knowledge, with more structured learning environments that allow students to practice problem-solving procedures step-by-step, to support students' acquisition of procedural knowledge. However, how best to combine both kinds of learning environments and thus both types of learning activities is an open question. We have developed a pedagogical intervention model that selects and sequences learning activities, exploratory learning activities and structured practice activities, that are appropriate for the individual learner. Technically, our intervention model is implemented as a rule-based system in a learning platform about fractions. The model's decision-making process relies on the detection of each individual student's level of challenge (i.e. whether they were under-, appropriately or over-challenged by the previous learning activity). Thus, our model adapts flexibly to each individual student's needs and provides them with a unique sequence of learning activities. Our formative evaluation trials suggest that single components of the intervention model, such as the ELE, mostly achieve their aims. The interplay between the different components of the intervention model (i.e. the outcomes of sequencing and selecting exploratory and structured practice activities) is currently being evaluated.

1 Introduction

Exploratory Learning Environments (ELEs), that include intelligent support, facilitate constructivist learning by offering opportunities for student self-determined exploration of a virtual environment [1]. The exploration of an ELE allows for sense-making activities which in turn promote the student's conceptual knowledge [2]. However, when integrating ELEs into the classroom, conceptual knowledge alone is insufficient. We need to move beyond this and enable students to achieve *robust* knowledge. Robust knowledge is deep, connected and comprehensive knowledge about a domain that lasts over time, accelerates future learning, transfers easily to new situations and is thus a very desirable learning goal [2–4]. It consists of both conceptual knowledge (understanding 'why') *and* procedural knowledge (knowing 'how') [5]. Thus, in addition to exploratory learning opportunities, we also need to provide students with learning opportunities that foster procedural knowledge [5] – opportunities for practicing problem-solving procedures, in structured learning environments such as that offered by some Intelligent Tutoring Systems (ITSs) [2] [6].

While prior work in the learning sciences and educational technology has mostly focused on fostering *either* procedural knowledge with structured practice activities (SPA) within ITSs *or* conceptual knowledge with exploratory learning activities (ELA) within ELEs, we aim to extend the existing literature by combining both types of learning activities – exploratory and structured – in order to help students acquire robust knowledge. This novel approach, combining both types of learning activities in one learning environment, also exploits the fact that conceptual and procedural knowledge evolve both iteratively and simultaneously [5].

Here, we report on a pedagogical intervention model (Figure 1), that specifies how to intelligently combine and sequence both ELA and SPA in order to promote complete robust knowledge. In doing so, we followed a theory and a data driven approach and thus iteratively improved our pedagogical model [7]. For example, our pedagogical intervention model builds on the cognitive psychology literature and, as such, is domain-neutral and thus transferable to other domains. However, as learning always depends on a target domain, the model also builds on previous work in the field of mathematics education, particularly fractions learning. The intervention model focuses on the individual student's level of challenge (categorized as either under-, appropriately or over-challenged) and selects the next learning activity accordingly. The model further specifies when students should receive cognitive support, so called task-dependent-support (TDS) , and emotional support, so called task-independent-support (TIS) [8]. The technical implementation of the intervention model is based on a rule-based system that, in order to determine each individual student's level of challenge, evaluates various input indicators (for example the student's response to the activity and the amount of feedback the system has provided).

A speech-enabled learning platform about fractions represents our intervention model and is embedded in the larger context of the 7^{th} grant European research project "iTalk2Learn". In the following sections, we explain the rationale behind the intervention model in more detail, in particular describing how we determine each student's level of challenge, and we finish by discussing future steps.

2 The pedagogical intervention model

When combining ELA and SPA, the first question we have to address is which should come first? We argue that students should first start with an ELA rather than an SPA. The benefits of beginning with an ELA are evident in findings from Kapur [9]. He was able to show that students who started with an ill-structured task (*cf.* ELA) and continued with a well-structured task (*cf.* SPA) gained significantly more conceptual knowledge than students learning in the reverse order. This research was extended by Kapur in his work on Productive Failure [10] which replicated the finding that exploring concepts first fosters conceptual knowledge without hampering the acquisition of procedural knowledge. The choice to start with an ELA was also rooted in a domain-specific reason. From more than 20 years of research, the Rational Number Project [11] elicited four essential beliefs about how best to support students learning fractions [12]. One of these essential beliefs is that "teaching materials for fractions should focus on the development of conceptual knowledge prior to formal work with symbols and algorithms" [13].

The next question to be addressed when combining ELAs and SPAs is what activity comes after the initial ELA? The answer depends on the individual student's level of challenge. Students who are over-challenged with the initial ELA should continue with another less challenging ELA, in order to prevent them applying rules without prior reasoning [14]. On the other hand, students who are under-challenged should be given a more challenging ELA, in order to extend their learning. Finally, for students who are appropriately challenged by the ELA, switching from the exploratory to a structured activity is useful because the acquisition of conceptual and procedural knowledge mutually depend upon each other: changes in one type of knowledge lead to changes in the other type of knowledge which in turn lead to changes in the first type [5]. For example, when a student is appropriately challenged by an ELA, an SPA that is mapped to the ELA allows the student to elaborate and consolidate the conceptual knowledge that was acquired during the ELA.

A third question to be addressed is once a student has engaged with a SPA, what activity comes next? In light of ACT-R theory [15] and the power law of practice [16] students should be provided with more than a single SPA because they need sufficient practice in order to become fluent in the application of a problem-solving procedure. Accordingly, the student should engage with more than a single SPA. In addition to providing students with opportunities to become fluent with a given procedure, we also aim to facilitate students' flexible retrieval of different procedures by providing them with interleaved practice of SPAs, rather than simple blocked practice [17, 18]. However, once the student has become fluent with a given procedure, then additional practice does not lead to better learning [17]. Therefore, students are switched back to the ELE. In this way, the student starts a new learning cycle, which (in the context of our project) is embedded in a particular coarse grain goal of fractions learning (e.g. equivalence of fractions). Here again, depending on the student's level of challenge, the new learning cycle focuses either on the same coarse grain goal, and thus provides the student with additional learning opportunities for that goal, or moves to another coarse grain goal (e.g. adding fractions).

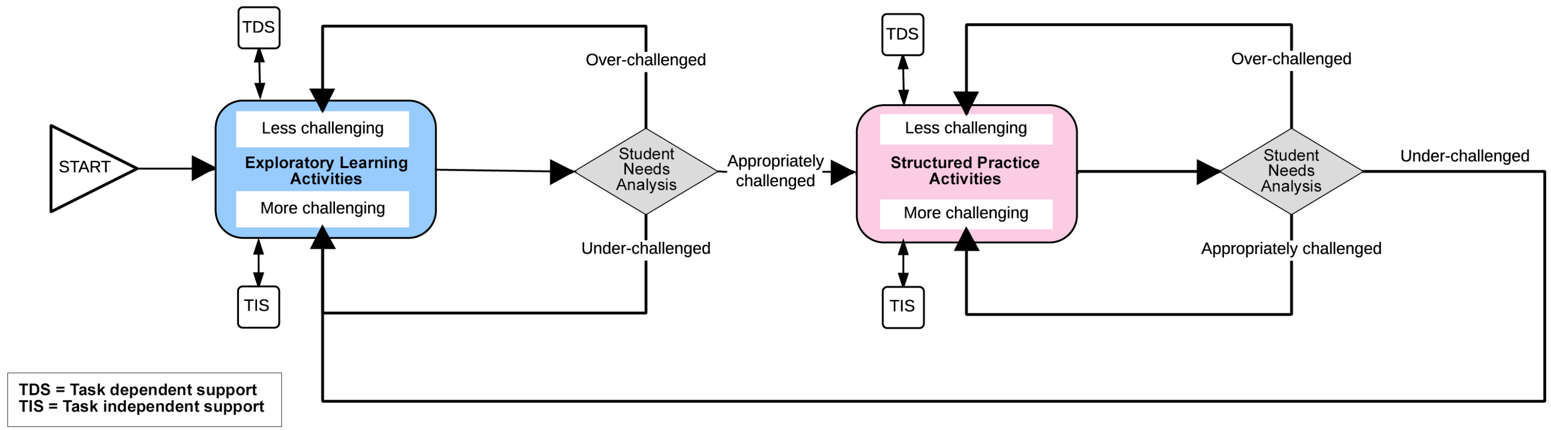

Figure 1: The pedagogical intervention model.

3 Determining a student's level of challenge

Determining a student's current level of challenge is a complex affair, because it is a function of characteristics both of the student *and* of the activity. For example, an ELA is likely to be less challenging for a student with high prior knowledge than for another student with low prior knowledge. Based on our pedagogical intervention model and a student model (i.e. considering the various input variables) the analytical engine (that we call the Students Needs Analysis or SNA) determines the student's level of challenge and thus the learner's appropriate next activity (i.e. output decision). For example, the SNA draws on the student's response to previous activities and to the current activity (using as a proxy the amount of task-dependent support, TDS [19], and the amount of task-independent support, TIS [8], delivered by the system), and the affective state inferred from the student's speech. Combining all these various inputs, each of which is assigned a weighting based on expert pedagogy, provides the SNA with a level of redundancy: a decision about the next appropriate activity can still be reached even if one of the inputs does not give any useful information or gives contradictory information.

3.1 Student Needs Analysis for exploratory learning activities

After each ELA, the SNA determines whether the student was under-, appropriately or over-challenged, based on the following input variables:

- the student's response to the current activity (using as a proxy the amount of TDS and TIS delivered by the system);
- the student's affect state inferred from prosodic cues in the student's speech;
- the student's affect state inferred from their screen and mouse behavior.

Based on these data, the SNA makes an output decision, selecting the next activity that is appropriate for the learner. If, for example, the system has had to deliver a large amount of TDS and the student's affective state has been calculated as *frustrated*, the SNA will determine that the student was over-challenged by the ELA and will sequence to a less challenging ELA. If, on the other hand, few TDS prompts have been delivered and the student's affect is inferred from speech to be *bored*, the SNA will determine that the student was under-challenged by the ELA and will sequence to a more challenging ELA.

Finally, if the SNA infers the student is appropriately challenged (for example, if there has been a minimal number of TDS and the affect has been categorized as *enjoyment*), the SNA switches to the structured practice environment. To ensure that students are provided opportunities to build upon and consolidate their conceptual knowledge, by applying it during structured practice, the SPA are mapped as closely as possible to the just-explored ELA. The close mapping of activities also aims to keep the individual student in their zone of proximal development, that is "the distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under adult [or an Intelligent Tutor's] guidance, or in collaboration with more capable peers" [20].

3.2 Student Needs Analysis for structured practice activities

After students have completed a SPA, the SNA determines what the next activity should be based on the following input indicators (a future implementation will also take account of the number of SPAs the student has completed and the time taken):

- performance prediction, based on a machine-learning model that uses a student's past activity performance to predict future activity performance [21];
- the student's affect state inferred from prosodic cues;
- the TIS previously delivered.

Here, again, the SNA determines whether the student is under-, appropriately or over-challenged. If the SNA detects that a student was over-challenged by a SPA and the student's affect is categorized as *frustrated*, the SNA will deliver a less challenging SPA. By providing over-challenged students with a less challenging SPA we aim to enable the student to become fluent with a less challenging procedure, before re-exposing him to the more challenging procedure that they had not managed before. On the other hand, if the SNA detects that the student is appropriately challenged, he will be assigned a more challenging SPA. A machine-learning-based performance prediction model is used to determine how challenging activities are to the student. It takes into account data about the student's performance on previous tasks and data from other students working on these tasks from a historic dataset. Finally, if the SNA detects that the student is under-challenged, the SNA will switch back to the ELE and will assign a new ELA that is more challenging than the last ELA that they explored.

4 Summary and outlook

Our intervention model, currently implemented within the context of learning fractions, combines exploratory learning activities (ELA) with structured practice activities (SPA) according to each individual student's level of challenge, in order to achieve robust knowledge. In addition to the adaptive selection of the next activity, our intervention model also provides adaptive support in the form of TDS and TIS during each learning activity. Accordingly, students are provided with both cognitive and emotional support as they learn about fractions. Although our intervention model evolved within the domain of fractions learning, it is transferable to other domains as the rationale behind the intervention model is domain-neutral.

Repeated formative evaluation trials across the UK and Germany have tested the effectiveness of all the separate components of the intervention model. For example, various Wizard-of-Oz studies have delivered first empirical evidence that our ELE and its TDS supports students' exploratory behavior and fosters their conceptual understanding of fractions. Meanwhile, the interplay between different components of the intervention model is currently being evaluated. To test the effectiveness of the intervention model we have created different versions of our learning platform. For example, in two quasi-experimental studies in the UK and Germany, we are comparing a full version of the learning platform representing our intervention model with a version that is without the ELE (but has all the other components). We expect differential effects in terms of students' knowledge acquisition (full version, complete robust knowledge, vs. the version without the ELE, procedural knowledge only) and user experiences. The initial results of these evaluation studies will be presented during the AIED workshop.

Once the learning platform is evaluated we will intensify our effort to facilitate the use of the platform for teachers by providing guidelines about how best to prepare for students' interaction with the platform. Additionally, for when working with the platform in class, we aim to provide teachers with a tool (e.g., a teacher dashboard) which will allow them to monitor individual student's use of the learning platform [22]. A further promising approach would be to enable students to learn collaboratively with the platform, as collaborative learning might further support students exploratory behavior and hence additionally support students' learning. From a more technical perspective, our next step is to develop a Bayesian network that is able to predict more precisely the learner-appropriate next activity. However, this first requires the collection of training data for the network from our current rule-based implementation of the SNA.

Acknowledgments. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 318051 - iTalk2Learn project. This publication reflects only the authors' views and the Union is not liable for any use that may be made of the information contained therein.

References

1. Noss, R. & Hoyles, C. Windows on Mathematical Meanings: Learning Cultures and Computers. Dordrect: Kluwer Academic Publishers. (1996)
2. Koedinger, K.R., Corbett, A.T., Perfetti, C.: The knowledge-learning-instruction framework: bridging the science-practice chasm to enhance robust student learning. Cognitive science 36, 757–798 (2012)
3. Richey, J.E., Nokes-Malach, T.J.: Comparing Four Instructional Techniques for Promoting Robust Knowledge. Educ Psychol Rev 27, 181–218 (2015)
4. Koedinger, K.R., Aleven, V.: Exploring the Assistance Dilemma in Experiments with Cognitive Tutors. Educ Psychol Rev 19, 239–264 (2007)
5. Rittle-Johnson, B., Siegler, R.S., Alibali, M.W.: Developing conceptual understanding and procedural skill in mathematics: An iterative process. Journal of Educational Psychology 93, 346–362 (2001)

6. van Lehn, K.: The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. Educational Psychologist 4, 197–211 (2011)
7. Cobb, P., Confrey, J., diSessa, A., Lehrer, R., Schauble, L.: Design Experiments in Educational Research. Educational Researcher 32, 9–13 (2003)
8. Grawemeyer, B., Holmes, W., Gutiérrez-Santos, S., Hansen, A., Loibl, K., Mavrikis, M. and Brdiczka, O.: Light-Bulb Moment? Towards Adaptive Presentation of Feedback based on Students' Affective State // IUI '15. Proceedings of the 20th International Conference on Intelligent User Interfaces ; March 29 - April 1, 2015, Atlanta, Georgia, USA, 400–404 (2014)
9. Kapur, M.: Productive Failure. Cognition and Instruction 26, 379–424 (2008)
10. Kapur, M.: Productive failure in learning the concept of variance. Instr Sci 40, 651–672 (2012)
11. Cramer, K., Behr, M., T., P., & Lesh, R. (1997). Rational Number Project: Fraction Lessons for the Middle Grades - Level 1. Dubuque Iowa: Kendall/Hunt Publishing Co. (1997)
12. Cramer, K.A., Post, T.R., delMas, R.C.: Initial Fraction Learning by Fourth- and Fifth-Grade Students: A Comparison of the Effects of Using Commercial Curricula with the Effects of Using the Rational Number Project Curriculum. Journal for Research in Mathematics Education 33, 111 (2002)
13. Cramer, K., & Henry, A. (2002). Using Manipulative Models to Build Number Sense for Addition of Fractions. In B.H. Littwiller & G.W. Bright (Eds.), Making Sense of Fractions, Ratios and Proportions: 2002 Yearbook (pp. 41-48). Reston, VA: National Council of Teachers of Mathematics. (2002)
14. Skemp, R. (1976). Relational Understanding and Instrumental Understanding Mathematics Teaching, 77, 20–26. (1976)
15. Anderson, J.R., Lebiere, C.: The atomic components of thought. Lawrence Erlbaum Associates., Mahwah, NJ (1998)
16. Newell, A.Hillsdale, NJ: Erlbaum & Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Ed.), Cognitive skills and their acquisition (pp. 1-55). (1981)
17. Rohrer, D., Taylor, K.: The effects of overlearning and distributed practise on the retention of mathematics knowledge. Appl. Cognit. Psychol. 20, 1209–1224 (2006)
18. Rau, M.A., Aleven, V., Rummel, N.: Interleaved practice in multi-dimensional learning tasks: Which dimension should we interleave? Learning and Instruction 23, 98–114 (2013)
19. Holmes, W., Mavrikis, M. Hansen, A., Grawemeyer, B.: Purpose and Level of Feedback in an Exploratory Learning Environment for Fractions. In: Proceedings of AIED (2015)
20. Vygotskiĭ, L.S., Cole, M.: Mind in society. The development of higher psychological processes. Harvard University Press, Cambridge (1978)
21. Schatten, C., Schmidt-Thieme, L.: Adaptive Content Sequencing without Domain Information. In: Proceedings of the 6th International Conference on Computer Supported Education (www.csedu.org), Springer (2014)

22. Dillenbourg, P., Jerman, P.: Technology for classroom orchestration. In: Khine, M.S. , Saleh, I.M. (ed.) New science of learning: Cognition, computers and collaboration in education, pp. 525–552. Springer (2010)

Adapting Collaboratively by Ranking Solution Difficulty: an Appraisal of the Teacher-Learner Dynamics in an Exploratory Environment

Rômulo C. Silva[1,2], Alexandre I. Direne[2], Diego Marczal[3], Paulo R. B. Guimarães[2], Ângelo S. Cabral[2], and Bruno F. Camargo[2]

[1] Western University of Paraná (UNIOESTE)
[2] Federal University of Paraná
[3] Federal Technological University of Paraná
{romulocesarsilva,dmarczal,guimaraes.prb,angeloscabral}@gmail.com
alexd@inf.ufpr.br
brunofilla_camargo@hotmail.com

Abstract. The work approaches theoretical and implementation issues of a framework aimed at supporting human knowledge acquisition of mathematical concepts. We argue that the problem solving tasks to be carried out by a learner should be ordered according to the matching of two parameters: (1) human skill level and (2) solution difficulty. Both are formally defined here as algebraic expressions based on fundamental principles derived from extensive consultations with experts in pedagogy and cognition. Our general definition of skill level is a rating-based measure that resembles the ones of game mastery scales. Likewise, the solution difficulty includes valuations based on a calibration method that computes mistakes and successes of learners' attempts to deal with the problem. The framework is instantiated by implemented software tools for the domain of logarithmic properties. Finally, we draw conclusions about the suitability of the claims based on a four-highschool-class experiment.

Keywords: rating, exercises calibration, Intelligent Tutoring Systems

1 Introduction

The student's expertise is usually developed by solving exercises that require a set of assessed skills. This is done in both conventional education schools and when applying advanced learning technologies, such as Intelligent Tutoring Systems (ITS). Normally, human teachers detect students' misconceptions when marking tests and exercises. Depending on how much the answer of a question departs from its correct version, two students that missed the same question could be scored different grades for that specific question.

Another aspect that can be used to compose the score is how difficult the question is. The difficulty degree of a question can be measured by the number of students that have skipped or made a mistake in that question. Thus, a student who finds the correct answer of a question that many missed, probably has more skills than others and the score should reflect that. Conversely, a student who makes a mistake in a question that many were successful to answer, might possess fewer skills. Therefore, when posing questions to a student, it's desirable that an ITS calibrates the difficulties of such questions properly in order to match them against the expertise level of the student.

The student models have become a key element in ITS, supporting the development of individual help and detecting off-task behaviour [1]. The more recent approaches of student displacement behaviour from what is expected are influenced by the other students' behaviour. In this sense, a larger sampling of learners should provide better automatic assessments of a specific learner.

In the construction of student models, an important issue is weather just one or multiple skills will be considered. Some of the proposed models are based on the IRT (Item Response Theory), which is a classical model in psicometrics that assumes that success in every item of a test is determinated by one ability, named θ, referred to as latent trait.

Another desirable aspect in ITS is predicting or prospecting if a learner will be able to answer a question correctly or not before it is actually showed to him or her. This feature allows the exercises to be presented according to the student's skills or rating.

2 Literature Review

Champaign and Cohen propose an algorithm [3] for content sequencing that selects the appropriate learning object to present to a student, based on previous learning experiences of like-minded users. The granularity of sequencing is on the LO level, not exercises or issues. A limitation of the work is that the algorithm was validated only by using simulated students.

Ravi and Sosnovsky [14] propose a calibration method for solution difficulty in ITS based on applying data mining techniques to a student's interaction log. Using the classical bayesian Knowledge Tracing (KT) method [5], the probability that a student has acquired a skill is calculated on the basis of a tentative sequence of exercises for which the soluctions involve a given concept. The logged events are grouped by exercises and classified according to the student's skills. All the data generated by the process is then used to match the sigmoid curve of IRT to connect different students using the standard clustering algorithm k-means.

Schatten and Schmidt-Thieme [15] present the Vygotski Policy Sequencer (VPS), based on the concept of Zone of Proximal Development devised by Vygotski. In this approach, the matrix factorization, which is a method for predicting user rating, is combined with a sequencing policy. This is done in order to select at each time step the content according to the predicted score.

Clement *et al.* [4] propose two algorithms for the tutoring model of ITS. The first, named RiARiT (Right Activity at Right Time), is based on multi-arm bandit techniques [2] such that each activity involves different skills, referred to as

Knowledge Components (KCs). The student model is a generalization of the one used in the bayesian KT method, representing the student's competence level (c_i) by a Real number in the range [0..1]. Furthermore, a reward representing the learning progress is defined by the difference between required KC and c_i. The second algorithm, ZPDES (Zone of Proximal Development and Empirical Success) [4] is a modified version of RiARiT where the calculation of the reward is changed in order to remove the dependence of the student's estimated competence level. The reward becomes a measure of how the success rate is increasing, providing a more predictive choice of activities.

Guzmán and Conejo [10] propose a cognitive assessment model based on IRT for ITS that calibrates the items of a topic (or concept). The method of item calibration is based on the kernel smoothing statistical technique that requires a reduced number of prior students sessions compared to conventional methods. In their approach, each possible answer has a characteristic curve that expresses the probability that a student with a certain knowledge level will more than likely select this answer.

There are several works about rating prediction techniques. Desmarais *et al.* [7] presented a comparative study between different linear models of student skill based on matrix factorization, IRT model and the k-nearest-neighbours approach. The linear models based on matrix factorization make predictions using a subset of the observed performance data for each student to predict the remaining subset, and measure the prediction accuracy. For other works, see [9], [6] and [16].

3 Automatic Calculation of Rating

Rating systems are frequently used in games to measure the players skills and to rank them. Usually, the rating is a number in a range $[minRank, maxRank]$ such that it is very unlikely that a player falls on the extremes. Inspired by game rating systems and taking the performance of other learners, this study proposes Equation 1 to assess iteratively a student's ability.

The following guidelines were adopted:*(1)* each question is scored a difficulty degree with a value in the range [0..10] and the student is rated in the range [1..10] to express his or her expertise level in the subject matter;*(2)* the easier the question, the greater the likelyhood that students will answer it correctly (in this case, a student's rating should have just a small increase if he or she enters the correct answer and should have a large decrease in the case of failure);*(3)* students that are successful in the first attempt to solve a question are scored a higher increment in their expertise level compared to those who need several attempts;*(4)* skipped questions are considered wrong.

Consider Equation 1.The details of its parameters are as follows:

$$R_J^q = R_J^{q-1} + Ak_1\alpha(10 - \frac{9T_J^q}{T_{med}^q}) - Ek_2\beta \times 10\frac{T_J^q}{T_{med}^q} \tag{1}$$

- R_J^q: student J's rating after answering question q;

- R_J^{q-1}: previous student J's rating. $R_J^0 = 5.5$ (initial rating);
- $A = 1$ and $E = 0$ if the student is successful in answering q, otherwise $A = 0$ and $E = 1$;
- T_J^q: number of unsuccessful attempts of student J to answer question q;
- T_{med}^q: median of wrong attempts on question q during classroom time;
- N_a^q: number of students that were successful in answering question q;
- N_e^q: number of students that were unsuccessful in answering question q;
- $\alpha = \frac{1}{N_a^q}$: weight factor to increase rating;
- $\beta = \frac{1}{N_e^q}$: weight factor to decrease rating;
- k_1 and k_2: multiplier factors of rating increase and decrease, respectively, calculated by $k_1 = 1 - \frac{R_J^{q-1}}{10}$ and $k_2 = \frac{R_J^{q-1}-1}{10}$.

Furthermore, $10 - \frac{9T_J^q}{T_{med}^q}$ and $10\frac{T_J^q}{T_{med}^q}$ represent the score of student J in question q in case the answer is correct and incorrect, respectively. There is no limit to the number of attempts T_J^q a student can make to answer a question. However, if there are more than 10 trials, then 10 is taken as the maximum value for calculation purposes. Factors k_1 and k_2 avoid results of the expression in Equation 1 to reach upper and lower bounds of the range [1..10].

Using only the number of attempts and considering that the student usually tries until he or she gets the correct answer, the difficulty degree of a question q can be defined by Equation 2 and its parameters as follows:

$$D^q = \frac{\sum_{J=0}^{J=n} T_J^q}{N_e^q + N_a^q} \tag{2}$$

- D^q: difficulty degree of the question q after an exercise session;
- T_J^q: number of unsuccessful attempts of student J to answer question q. If the number of attempts is greater than 10 trials, then 10 is taken as T_J^q;
- N_e^q and N_a^q are the same as in Equation 1

4 The ADAPTFARMA environment

The ADAPTFARMA (Adaptive Authoring Tool for Remediation of errors with Mobile Learning) prototype software tool is a modified version of FARMA[12], an authoring shell for building mathematical learning objects. In ADAPTFARMA, a learning object (LO) consists of a sequence of exercises following their introduction. The introduction is the theoretical part of a LO where concepts are defined through text, images, sounds and videos. The ADAPTFARMA implementation was carried out aiming its use on the web, either through personal computers or mobile devices.

To build an introduction and its corresponding exercise statements, ADAPTFARMA offers a WYSIWYG (What you See Is What You Get) interface, similar to those of highly interactive word processors. The teacher defines the number of questions related to each exercise. For each question, the teacher-author must set a reference solution, which is the correct response to the question. ADAPTFARMA allows arithmetic and algebraic expressions to be entered as the reference solution. Under the learner's functioning mode, the tool deals automatically with the equivalence between the learners response and the reference solution.

A feature of ADAPTFARMA is the capability of backtracking the teacher to the exact context in which the learner made a mistake. This gives the opportunity to the teacher to identify the wrong steps performed by the learner and, thus, deal with the causes of the error accordingly. In addition, ADAPTFARMA allows the teacher to view a learner's complete interaction with the tool in a chronological order, in the form of a timeline. The teacher can make a closer monitoring of problem solution from other classrooms, as long as system permission is given through the collaboration mechanisms.

Likewise, learners can backtrack to the context of any of their right or wrong answers in order to reflect about their own solution steps. Additionally, on the collaborative side, it is possible for the teacher to carry out a review of students' responses and then provide them with non-automatic feedback, which can be done by exchanging remote messages through the system.

5 Algorithm for Exercises Sequencing

An important aspect in ITS is how the exercises should be sequenced after they are calibrated in order to match them to the expertise level of the student. At the beginning, the system doesn't have any information about the student. We propose an algorithm for sequencing exercises to be shown in ascending order of difficulty, combined with a mechanism similar to numerical interpolation:

- a minimal sequence of exercises is defined such that always begins with the easiest exercise and finishes with the most difficult one;
- the intermediate level exercises in the minimal sequence are distributed evenly among the easiest and most difficult exercises such that the number of exercises is $\left\lceil \frac{n}{stepsize} \right\rceil$ where n is the total of exercises and the *stepsize* refers to the number of exercises that may be skipped when the student is successful. The *stepsize* can be set by the LO's author;
- the exercises are presented in the minimal sequence order;
- the number of attempts is limited to the average number of attempts obtained in the calibration phase. When the number of attempts is exceeded, the next exercise presented to the student is of a mid range difficulty considering the last exercise correctly answered and the current one.

For example, consider a LO with 30 exercises in ascending order of difficulty $[e_1, e_2, ..., e_{30}]$ and $stepsize = 4$. The minimal sequence of exercises will be $min\_seq =< e_1, e_5, e_9, e_{13}, e_{17}, e_{21}, e_{25}, e_{29}, e_{30} >$, and the exercises will be presented to the student in that order at first. For example, if the student misses e_9 until the attempts are over, then e_7 (of mid range difficulty between e_5 e e_9) is presented. Unlike the calibration phase, the student cannot skip exercises and if he/she continually misses the correct answer, the presentation becomes sequential.

6 Experiment

In order to evaluate the learning effectiveness of the four sequencing strategies, we carried out an experiment with four different classes of highschool students, aging fifteen to seventeen. The same LO about logarithms was applied to all four classes. It was created with the ADAPTFARMA enviroment to include thirty exercises. For each class, the LO was applied with a different sequencing method to order the exercises as follows:

- class A: random sequencing method (RSM);
- class B: teacher-defined sequencing method (TSM);
- class C: difficulty-biased sequencing method (DSM), where the difficulty degree was calculated by Equation 2 using outcome data from the calibration phase of class A;
- Class D: adaptive sequencing method (ASM), using the algorithm described in the previous section

The same pre- and post-tests were applied to all four classes. Students who did not participate in any step have been excluded from the analysis, resulting 119 participants. For the RSM, TSM and DSM methods, there was no limit to the solution attempts while in ASM, the average of attempts in class A was used. The Shapiro-Wilk test was applied to all samples to check for normality. Because only the DSM data passed the normality test (p-value = 0.0827), the pairwise T Student test was applied to it (p-value = 0.532). For the other three, the choice was the Wilcoxon test in order to evaluate the individual sequencing methods. The p-value of RSM, TSM and ASM were 0.0007, < 0.0001 and 0.0037, respectively. All methods, except for DSM, had a significant increase in scores.

The ANOVA method was applied to the pre-test data that showed normality whereas the Kruskal-Wallis, to the others, both to the post-test and to the average difference between pre- and post-tests. The results indicate that there is no significant difference among the four classes in the pre-test scores (p-value = 0.2539). However, there is significant difference in the post-test scores (p-value = 0.00579) and in the average difference between pre- and post-tests scores (p-value = 0.0307), suggesting that RSM, TSM and ASM led to better student performance than DSM. Besides, student performances among the three (RSM, TSM and ASM) were similar. Surprisingly, RSM led to the best performance while DSM, to the worst. This contradicts quite a large proportion of literature research on pedagogic practice, machine-led [8] or otherwise, for developing problem solving skills. Some reasons might explain such a phenomenon:

- the problem-statement ordering is a relevant issue that should be whatched more carefully to verify the influence of tacit knowledge contained in the textual organization of the statement;
- the lack of significant differences between RSM, TSM and ASM is also supported by evidence based on past research findings [11, 13];

– the DSM may have connected some sort of subject matters that caused an increase in the cognitive load, resulting in problem solutions that diverted from the correct ones;
– although most students have participated in the experiment, only the scores of pre- and post-tests accounted for the final student score in the official school records.

7 Conclusion and Future Work

Usually the student's expertise is developed by solving exercises that require a set of assessed skills, including ITS. We proposed an automatic rating system that can be used as an additional tool to assess students. Depending on the number of attempts and the difficulty degree of a question, students can get different scores for the same question.

Also, we proposed an algorithm, referred as ASM, for sequencing exercises that uses difficulty degree combined with a mechanism similar to numerical interpolation. It composes the ADAPTFARMA environment, a web authoring tool with WYSIWYG interface for creating and executing LOs. Taking advantage of it is very easy to change the strategy for exercises sequencing, we carried out a four-highschool-class experiment to test different sequences strategies: RSM, TSM, DSM and ASM. Only DSM had not a significant increase in the students' scores and the RSM had the best performance, demonstrating that problem-statement ordering is a relevant issue that should be researched more carefully in the near future. The ASM had also better performance compared to DSM.

Future research concentrates in adding new features to FARMA in two ways. Firstly, we are working in a deeper approach to user adaptation that includes more dimensions than just the matching between problem difficulty and student skill. One such new feature will be a function for generating problem statements based on teacher-defined problem statement parameters. Secondly, on the interface side, more interaction modes will be available to improve collaboration tasks for monitoring student performance progress.

References

1. Ryan S.J. D. Baker, Adam B. Goldstein, and Neil T. Heffernan. Detecting the Moment of Learning. *LNCS*, 6094(PART I):25–34, 2010. Springer-Verlag Berlin Heidelberg.
2. Sébastin Bubeck and Nicolò Cesa-Bianchi. Regret analysis of Stochastic and Non-stochastic Multi-armed Bandit Problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.
3. John Champaign and Robin Cohen. A Model for Content Sequencing in Intelligent Tutorign Systems Based on the Ecological Approach and Its Validation Through Simulated Students. pages 486–491. Association for the Advancement of Artificial Intelligence (AAAI), 2010.

4. Benjamin Clement, Didier Roy, and Pierre-Yves Oudeyer. Online Optimization of Teaching Sequences with Multi-Armed Bandits. In Pardos Z. Mavrikis M. McLaren B.M. Stamper, J., editor, *Proceedings of the 7th International Conference on Educational Data Mining*, pages 269–272, 2014.
5. Albert T. Corbett and John R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1994.
6. Maunendra Sankar Desarkar and Sudeshna Sarkar. Rating prediction using preference relations based matrix factorization. In Kalina Yacef Eelco Herder and Stephan Weibelzahl, editors, *Workshop and Poster Proceedings of the 20th Conference on User Modeling, Adaptation, and Personalization (UMAP2012)*, volume 872. http://http://ceur-ws.org/, Jul 2012.
7. Michel C. Desmarais, Rhouma Naceur, and Behzad Beheshti. Linear models of student skills for static data. In Kalina Yacef Eelco Herder and Stephan Weibelzahl, editors, *Workshop and Poster Proceedings of the 20th Conference on User Modeling, Adaptation, and Personalization (UMAP2012)*, volume 872. http://http://ceur-ws.org/, Jul 2012.
8. Alexandre Direne. Authoring intelligent systems for teaching visual concepts. *International Journal of Artificial Intelligence in Education*, 1(4):3–14, 1990.
9. Lucas Drumond, Nguyen Thai-Nghe, Tomáš Horváth, and Lars Schmidt-Thieme. Factorization techniques for student performance classification and ranking. In Kalina Yacef Eelco Herder and Stephan Weibelzahl, editors, *Workshop and Poster Proceedings of the 20th Conference on User Modeling, Adaptation, and Personalization (UMAP2012)*, volume 872. http://http://ceur-ws.org/, Jul 2012.
10. Eduardo Guzmán and Ricardo Conejo. Towards efficient item calibration in adaptive testing. In Liliana Ardissono, Paul Brna, and Antonija Mitrovic, editors, *User Modeling 2005*, volume 3538 of *Lecture Notes in Computer Science*, pages 402–406. Springer Berlin Heidelberg, 2005.
11. N. Major and H. Reichgelt. COCA: A shell for intelligent tutoring systems. In *Proc. of the International Conference on Intelligent Tutoring Systems (ITS92)*, pages 523–530. Springer, 1992.
12. Diego Marczal and Alexandre Direne. Farma: Uma ferramenta de autoria para objetos de aprendizagem de conceitos matemáticos. In *Anais do Simpósio Brasileiro de Informática na Educação*, volume 23, 2012.
13. Antonija Mitrovic. An intelligent SQL tutor on the web. *International Journal of Artificial Intelligence in Education*, 13(3):173–197, 2003.
14. Gautham Adithya Ravi and Sergey Sosnovsky. Exercise difficulty Calibration Based on Student Log Mining. In F. Mödritscher, V. Luengo, E. Lai-Chong Law, and U. Hoppe, editors, *Proceedings of DAILE'13: Workshop on Data Analysis and Interpretation for Learning Environments*, Villard-de-Lans (France), Janeiro 2013.
15. Carlotta Schatten and Lars Schmidt-Thieme. Adaptive Content Sequencing without Domain Information. *6th International Conference on Computer based Education*, April 2014.
16. Avi Segal, Ziv Katzir, Kobi Gal, Guy Shani, and Bracha Shapira. EduRank: A Collaborative Filtering Approach to Personalization in E-learning. In Pardos Z. Mavrikis M. McLaren B.M. Stamper, J., editor, *Proceedings of the 7th International Conference on Educational Data Mining*, pages 68–75, 2014.

Towards Using Coherence Analysis to Scaffold Students in Open-Ended Learning Environments

James. R. Segedy & 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,john.s.kinnebrew,gautam.biswas}@vanderbilt.edu

Abstract. Scaffolding students in open-ended learning environments (OELEs) is a difficult challenge. The open-ended nature of OELEs allows students to simultaneously pursue, modify, and abandon any of a large number of both short-term and long-term approaches to completing their tasks. To overcome these challenges, we have recently developed *coherence analysis*, which focuses on students' ability to *interpret* and *apply* the information available in the OELE. This approach has yielded valuable dividends: by characterizing students according to the coherence of their behavior, teachers and researchers have access to easily-calculated, intuitive, and actionable measures of the *quality* of students' problem-solving processes. The next step in this line of research is to develop a framework for using coherence analysis to adaptively scaffold students in OELEs. In this paper, we present our initial ideas for this work and propose guidelines for the construction of a scaffolding framework.

Keywords: Open-ended learning environments, metacognition, coherence analysis, scaffold

1 Introduction

Open-ended computer-based learning environments (OELEs) [1-2] are learner-centered; they present students with a challenging problem-solving task, information resources, and tools for completing the task. Students must use the resources and tools to construct and verify problem solutions, and in this process learn about the problem domain and develop their general problem-solving abilities. In OELEs, students have to distribute their time and effort between exploring and organizing their knowledge, creating and testing hypotheses, and using their learned knowledge to create solutions. Since there are no prescribed solution steps, students may have to discover the solution process over several hours. For example, learners may be given the following:

> *Use the provided simulation software to investigate which properties relate to the distance that a ball will travel when rolled down a ramp, and then use what you learn to design a wheelchair ramp for a community center.*

Whereas OELEs support a constructivist approach to learning, they also place significant cognitive demands on learners. To solve problems, students must simultaneously wrestle with their emerging understanding of complex topics, develop and utilize skills to support their learning, and employ *self-regulated learning* (SRL) processes to manage the open-ended nature of the task. SRL is a theory of learning that describes how learners actively set goals, create plans for achieving those goals, continually monitor their progress, and revise their plans when necessary to continue to make progress [3]. As such, OELEs can *prepare students for future learning* [4] by developing their ability to independently investigate and develop solutions for complex open-ended problems.

However, research with OELEs has produced mixed results. While some students with higher levels of prior knowledge and SRL skills show large learning gains as a result of using OELEs, many of their less capable counterparts experience significant confusion and frustration [5-7]. Research examining the activity patterns of those students indicates that they typically make ineffective, suboptimal learning choices when they independently work toward completing open-ended tasks [7-10].

The strong self-regulatory component of OELEs makes them an ideal environment for studying SRL. The open-ended nature of the environment forces students to make choices about how to proceed, and these choices reveal information about students' understanding of: (i) the problem domain; (ii) the problem-solving task; and (iii) strategies for solving the problem. By studying these choices, we can gain a better understanding of how students regulate their learning and how best to design scaffolds to support students who struggle to succeed.

Recently, we have introduced *coherence analysis* (CA) [11], a technique for studying students' problem-solving behaviors in OELEs. CA analyzes learners' behaviors in terms of their demonstrated ability to seek out, interpret, and apply information encountered while working in the OELE. By characterizing behaviors in this manner, CA provides insight into students' problem-solving strategies as well as the extent to which they understand the nuances of the learning and problem solving tasks they are currently completing.

In this paper, we present an overview of our findings with coherence analysis as applied to the *Betty's Brain* OELE (REF) and present our plans on extending this research. Our goal with CA is to empower both human and virtual tutors to more powerfully support students as they learn complex open-ended problem solving.

2 Betty's Brain

Betty's Brain [11] presents the task of teaching a virtual agent, Betty, about a science phenomenon (e.g., climate change) by constructing a causal map that represents that phenomenon as a set of entities connected by directed links representing causal relationships. Once taught, Betty can use the map to answer causal questions. The goal for students is to construct a causal map that matches an expert model of the domain.

In *Betty's Brain*, students acquire domain knowledge by reading resources that include descriptions of scientific processes (*e.g.*, shivering) and information pertaining to each concept that appears in the expert map (*e.g.*, friction). As students read, they need to identify causal relations such as "*skeletal muscle contractions create friction in the body.*" Students can then apply this information by adding the entities to the map and creating a causal link between them (which "teaches" the information to Betty). Learners are provided with the list of concepts, and link definitions may be either increase (+) or decrease (-).

Learners can assess their causal map by asking Betty to answer questions and explain her answers. To answer questions, Betty applies qualitative reasoning to the causal map (*e.g.*, *the question said that the hypothalamus response increases. This causes skin contraction to increase. The increase in skin contraction causes…*). After Betty answers a question, learners can ask Mr. Davis, another pedagogical agent that serves as the student's mentor, to evaluate her answer. If Betty's answer and explanation match the expert model (*i.e.*, in answering the question, both maps utilize the same causal links), then Betty's answer is correct.

Learners can also have Betty take *quizzes* (by answering sets of questions). Quiz questions are selected dynamically by comparing Betty's current causal map to the expert map such that a portion of the chosen questions, in proportion to the completeness of the current map, will be answered correctly by Betty. The rest of her quiz answers will be incorrect or incomplete, helping the student identify areas for correction or further exploration. When Betty answers a question correctly, students know that the links she used to answer that question are correct. Otherwise, they know that at least one of the links she used to answer the question is incorrect. Students may keep track of correct links by annotating them as such.

3 Coherence Analysis

The Coherence Analysis (CA) approach analyzes learners' behaviors by combining information from sequences of student actions to produce measures of *action coherence*. CA interprets students' behaviors in terms of the information they encounter in the OELE and whether or not this information is utilized during subsequent actions. When students take actions that put them into contact with information that can help them improve their current solution, they have *generated potential* that should *motivate future actions*. The assumption is that if students can recognize relevant information in the resources and quiz results, then they should act on that information. If they do not act on information that they encountered previously, CA assumes that they did not recognize or understand the relevance of that information. This may stem from incomplete or incorrect understanding of the domain under study, the learning task, and/or strategies for completing the learning task. Additionally, when students add to or edit their problem solution when they have not encountered any information that could motivate that edit, CA assumes that they are guessing[1]. These two notions come together in the definition of action coherence:

---

1 Students may be applying their prior knowledge, but the assumption is that they are novices to the domain and should verify their prior knowledge during learning.

*Two ordered actions ($x \rightarrow y$) taken by a student in an OELE are **action coherent** if the second action, y, is based on information generated by the first action, x. In this case, x provides **support** for y, and y is **supported** by x. Should a learner execute x without subsequently executing y, the learner has created **unused potential** in relation to y. Note that actions x and y need not be consecutive.*

CA assumes that learners with higher levels of action coherence possess stronger metacognitive knowledge and task understanding. Thus, these learners will perform a larger proportion of supported actions and take advantage of a larger proportion of the potential that their actions generate. In the analyses performed to date, we have incorporated the following coherence relations:

- Accessing a resource page that discusses two concepts *provides support for* adding, removing, or editing a causal link that connects those concepts.
- Viewing assessment information (usually quiz results) that proves that a specific causal link is correct *provides support for* adding that causal link to the map (if not present) and annotating it as being correct (if not annotated).
- Viewing assessment information (usually quiz results) that proves that a specific causal link is incorrect *provides support for* deleting it from the map (if present).

Using these coherence relations, we derived six primary measures describing students' problem solving processes:

1. *Edit Frequency*: The number of causal link edits and annotations made by the student per minute on the system.
2. *Unsupported edit percentage*: the percentage of causal link edits and annotations not supported by information encountered within 5 minutes of the edit/annotation.
3. *Information viewing time*: the amount of time spent viewing either the science resources or Betty's graded answers. *Information viewing percentage* is the percentage of the student's time on the system classified as *information viewing time*.
4. *Potential generation time*: the amount of *information viewing time* spent viewing information that could support causal map edits that would improve the map. To calculate this, we annotated each hypertext resource page with information about the concepts and links discussed on that page. *Potential generation percentage* is the percentage of *information viewing time* classified as *potential generation time*.
5. *Used potential time*: the amount of *potential generation time* associated with information viewing that both occurs within a prior five minute window of and also supports an ensuing causal map edit. *Used potential percentage* is the percentage of *potential generation time* classified as *used potential time*.
6. *Disengaged time*: the sum of all periods of time, at least five minutes long, during which the student neither viewed a source of information for at least 30 seconds nor edited the map. *Disengaged percentage* is the percentage of the student's time on the system classified as disengaged time.

Metrics one and two capture the quantity and quality of a student's causal link edits and annotations, where supported edits and annotations are considered to be of higher quality. Metrics three, four, and five capture the quantity and quality of the student's time viewing either the resources or Betty's graded answers. These metrics speak to the student's ability to seek and identify information that may help them build or refine their map (potential generation percentage) and then utilize information from those pages in future map editing activities (used potential percentage). Metric 6 represents periods of time during which the learner is not measurably engaged with the system.

3.1 Summary of Findings with Coherence Analysis

Coherence analysis has proved to be a valuable tool for understanding how students learn as they solve open-ended problems. Thus far, we have investigated it with one group of 98 6^{th}-grade students (11 year olds). Thus, we interpret our findings with cautious optimism. We have identified the following relationships:

- *CA predicts learning and performance*: in general, students with higher levels of coherent behaviors have shown significantly higher levels of success in teaching Betty. Moreover, these learners have shown a better understanding of the science domain they were learning [11].
- *Prior skill levels predict CA*: students who were better able to identify causal links in abstract text passages (*e.g.*, A decrease in Ticks leads to an increase in Tacks) exhibited higher levels of coherence while using *Betty's Brain* [11].
- *CA identifies common problem solving profiles across students*: we clustered students by describing them with the six CA metrics described above, and we identified five common profiles among students: researchers and careful editors; strategic experimenters; confused guessers; disengaged students; engaged and efficient students. Interestingly, there were few differences in learning and performance among the clusters. Engaged and efficient students showed higher learning and performance than the other clusters, but there were not any other meaningful differences, suggesting that CA allows us to understand how different learning approaches lead to similar learning outcomes [11].
- *CA identifies common day-to-day problem solving profiles and transitions among them*: we clustered students as before, but this time the unit of analysis was a single day of using the system instead of the entire time using the system. We found a set of behavior profiles quite similar to those identified in the previous analysis. In analyzing day-to-day transitions, we found that many students performed fairly consistently while several other students performed inconsistently (that is, they have days of high coherence and days of low coherence). We also identified common transitions among days, which allowed us to find a potentially *at-risk* behavior profile. Students who behave like researchers and careful editors are far more likely than chance to transition to confused or disengaged behavior in subsequent days [12].

4 An Initial Coherence-Based Scaffolding Framework

Given the previous findings with CA, we aim to utilize the power of the analysis in real time as students use the system in order to detect non-coherent behavior, diagnose the cause of it, and take steps to support students in overcoming the difficulties they are experiencing. The core idea behind CA is that when students work in OELEs, they have two primary sets of tasks: *information seeking tasks* related to identifying and interpreting important information and *information application tasks* related to applying that information to improving the problem solution. All coherence metrics are based on identifying relationships between activities related to these two sets of tasks. By analyzing student behaviors with CA, we can identify problems related to information seeking and information application.

4.1 Diagnosing Problems with CA Metrics

The initial framework for diagnosing problems using CA metrics appears in Figure 1. This framework maps CA metrics to the problems they may indicate. For example, low levels of potential generation indicate that the learner is spending a large portion of their information viewing time on non-helpful information. This indicates that they may be struggling to identify relevant vs. non-relevant information in the environment. Problems with information seeking may also manifest as high levels of unused potential (*i.e.*, not applying viewed information), a high proportion of unsupported edits, and a low rate of editing the solution. Problems with information application are indicated by high unused potential and a low rate of editing the solution.

CA metrics may also be used to identify behaviors associated with effort avoidance. Specifically, low levels of information viewing, a low rate of editing the solution, a high unsupported edit percentage, and high levels of disengagement indicate that the learner may be purposefully avoiding effort. This may be due to a number of reasons, including low self-efficacy and low skill understandings.

Using this framework, our initial plan for using CA to scaffold students is as follows:

1. Observe the student for a period of time (*e.g.*, 10 minutes) and calculate their coherence metrics for that period. Identify any problematic behaviors (*e.g.*, high unused potential).
2. Form hypotheses about the sources of these behaviors. This involves looking at the combination of problematic behaviors observed and the student's previous activities in the system. For example, if the problematic behaviors are high unused potential and a low editing rate, the system may hypothesize that the student is struggling to apply information.

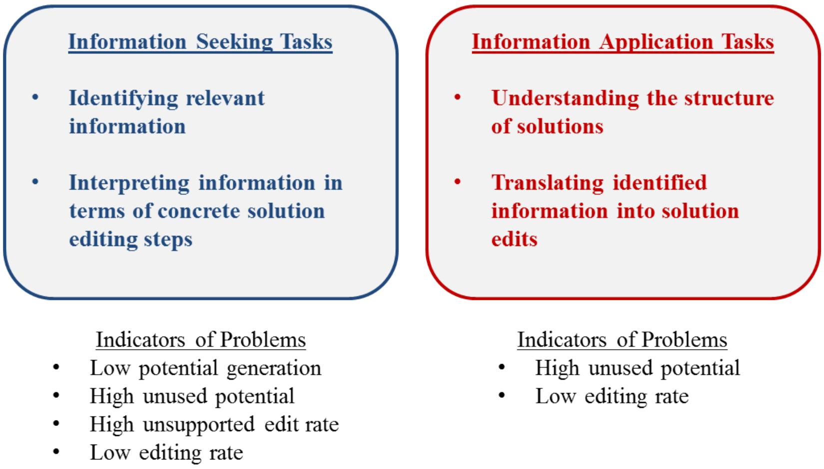

Fig. 1. Initial Problem Diagnosis Framework

3. Perform active diagnosis of the student to resolve competing hypotheses and gain additional information. For example, if the student has a high unsupported edit rate, this may be due to effort avoidance or a misunderstanding related to information seeking. The system can have the student answer questions and complete short problems in order to gain additional evidence as to which of these is the actual problem.
4. Once the system is confident that the student is struggling to understand something, it can use *guided practice scaffolds* [13] to help the student learn the knowledge and skills that they are missing or about which they are confused. Throughout guided practice, the system should provide encouragement, feedback, and scaffolding. It should also reinforce the relevance of the targeted knowledge and skills to the primary problem solving task, problem solving in general, and academic success.
5. If the system is confident that the student is exhibiting effort avoidance, then it should offer to help the student. If the behavior continues after the offer (and potential scaffolding related to that offer), then the system should provide guided practice scaffolds on the important knowledge and skills they need to understand to be successful. Hopefully, the student's abilities will improve during guided practice, and that will re-engage them with the learning task. As in the previous step, the system should provide the student with encouragement, feedback, and scaffolding and it should reinforce the relevance of the targeted knowledge and skills.

5 Conclusion

In this paper, we have provided an overview of *coherence analysis* (CA), an analysis approach that provides insight into how students behavior in open-ended computer-based learning environments (OELEs). Additionally, we have presented an initial scaffolding framework that describes how CA might be leveraged to provide adaptive scaffolds to students who are struggling. As we move forward, we will continue developing this scaffolding framework, build it into *Betty's Brain*, and test its effectiveness with students.

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

References

1. Land, S., Hannafin, M., Oliver, K.: Student-centered learning environments: Foundations, assumptions and design. In: D. Jonassen & S. Land (eds.) Theoretical Foundations of Learning Environments, pp. 3-25. Routledge, New York, NY (2012)
2. Segedy, J.R., Biswas, G., Sulcer, B.: A model-based behavior analysis approach for open-ended environments. The Journal of Educational Technology & Society, 17(1), 272-282 (2014)
3. Zimmerman, B., Schunk, D. (eds.): Handbook of Self-Regulation of Learning and Performance. Routledge , New York, NY (2011)
4. Bransford, J., Schwartz, D.: Rethinking transfer: A simple proposal with multiple implications. Review of Research in Education, 24(1), 61-101 (1999)
5. Azevedo, R., Witherspoon, A.: Self-regulated learning with hypermedia. In: Hacker, D., Dunloskey, J., Graesser, A. (eds.), Handbook of Metacognition in Education, pp. 319-339. Taylor and Francis (2009)
6. Hacker, D., Dunloskey, J., Graesser, A. (eds.), Handbook of Metacognition in Education, pp. 319-339. Taylor and Francis (2009)
7. Kinnebrew, J.S., Loretz, K., Biswas, G.: A contextualized, differential sequence mining method to derive students' learning behavior patterns. Journal of Educational Data Mining, 5(1), 190-219 (2013)
8. Land, S.M.: Cognitive requirements for learning with open-ended learning environments. Educational Technology Research and Development*,* 48, 61-78 (2000)
9. Mayer, R.E.: Should there be a three-strikes rule against pure discovery learning?, American Psychologist, 59, 14-19 (2004)
10. Sabourin, J., Shores, L., Mott, B., Lester, J.: Understanding and predicting student self-regulated learning strategies in game-based environments. International Journal of Artificial Intelligence in Education, 23, 94-114 (2013)
11. Segedy, J.R., Kinnebrew, J.S., Biswas, G.: Using coherence analysis to characterize self-regulated learning behaviours in open-ended learning environments. Journal of Learning Analytics (in press)
12. Segedy, J.R., Biswas, G.: Coherence over time: Understanding day-to-day changes in students' open-ended problem solving behaviors. In: Proceedings of the 17th International Conference on Artificial Intelligence in Education (in press)
13. Segedy, J., Biswas, G., Blackstock, E., Jenkins, A.: Guided skill practice as an adaptive scaffolding strategy in open-ended learning environments. In: Proceedings of the 16th International Conference on Artificial Intelligence in Education. pp. 532-541. Springer (2013)

Design Strategies for developing a Visual Platform for Physical Computing with Mobile Tools for Project Documentation and Reflection

Daniel Spikol[1], Nils Ehrenberg[1], David Cuartielles[2], and Janosch Zbick[3]

[1] Malmö University, Malmö, 211 19 Sweden
daniel.spikol@mah.se
[2] Arduino Verkstad, Malmö, 211 19 Sweden
[3] Linnæus University, Växjö, 351 95 Sweden

Abstract. This poster discusses work on the design of a visual-based programming language for physical computing and mobile tools for the learners to actively document and reflect on their projects. These are parts of a European project that is investigating how to generate, analyze, use and provide feedback from analytics derived from hands-on learning activities. Our aim is to raise a discussion about how learning analytics, intelligence, and the role of learners' documenting their work can provide richer opportunities for supporting learning and teaching.

Keywords: learning analytics, human factors and interface design, prototyping

1 Introduction

Educators, researchers, business leaders, and politicians are working to initiate new modes of education to provide 21st Century skills that focus on the following: creativity, innovation, critical thinking, problem solving, communication, and collaboration [4]. Recently, researchers and practitioners have provided strong cases for the value of hands-on activities like digital fabrication than could be part of the toolbox to bring powerful ideas, literacies, and expressive tools to learners [1]. This poster presents on-going work in the Practice-based Experiential Learning Analytics Research And Support project (PELARS) that aims to generate, analyze, use and provide feedback for analytics derived from hands-on these project-based learning activities. The focus of the PELARS project activities is on learning and making things with physical computing that provide learners with opportunities to build and experiment with tangible technologies and digital fabrication. One of the key research aims of the PELARS project can be summarised as: *How can physical learning environments that use hands-on digital fabrication technologies be better designed for ambient and active data collection for learning analytics?* The project addresses three different learning contexts (university interaction design, engineering courses, and high school science) across multiple settings in Europe. The goals of the project are first to define learning (skills, knowledge, competencies) that is developing, and how we can assess it in the frame of learning analytics. Then to determine what elements of this learning we can capture by designing the physical environment and activities around digital fabrication technologies. Then to identify what patterns of data we collect can tell us about learning, collaboration and how the system can help support the learning activities.

The PELARS project approach has been to develop an intelligent system for collecting activity data (moving image-based and embedded sensing) for diverse learning analytics (data-mining, reasoning, visualisation) with active user-generated material from practice-based and experiential activities. This rich range of data is used to create learning analytics tools for learners and teachers that range from assessment to exploring intelligent tutoring. The PELARS system carries forwards the ideas of knowledge communities and inquiry [7] and provide conceptualising, representing, and analysing distributed interaction [8]. However, there are multiple challenges for designing learning analytics and intelligent support for these types of tangible activities. Learning situations in these contexts include open-ended projects, small group work, and the use of physical computing components that require construction and programming. Therefore, these types of activities present difficulties for collecting meaningful data for learning analytics.

This poster specifically discusses our work on the development of a visually based programming platform for the physical computing hardware and the mobile tools for the learners to actively document and reflect on their projects. These two parts of the PELARS project provide opportunities for discussion on the relationships between intelligent support, active learner engagement, and analytics. Our aim for the workshop is to raise a discussion about how learning analytics, intelligence, and the role of learner documenting their work can provide richer opportunities for supporting learning.

2 Methodological Approach

The PELARS project has a design-centric approach that includes the use of low-fidelity prototyping and "wizard of oz" scenarios [5]. These methods that include paper prototypes and technology sketches to investigate how to find the best way to get the design right [2]. The goal of the two cases below is to investigate how we can better understand the needs of the users. The need to develop a visually based programming experience to support students and supply data for analysis and lack of student documentation were identified as challenges through literature and own contextual user research in the project.

For the visual programming platform, a kit was created that contained foam core versions of hardware blocks with strings and pins to act as the cables to connect them. A small magnetic board with paper-based magnets acted as the computer screen that represented what blocks were connected. A set of simple tasks were provided to pairs of testers (recruited students from Interaction Design and Computer Science) while one of the researchers acted as the computer in a "wizard of oz" scenario. Figure 1 illustrates how the students connected the hardware blocks of sensors and actuators (the paper blocks and strings) on the table and then the researcher put on a magnetic board (computer screen) the associated blocks (printed magnets). The researcher acted as the wizard representing the smart system that recognised which blocks the students had connected and represented the computer screen showing the visual programming interface. This prototyping system allowed the teams to discuss and adjust the inputs to generate the hypothetical outcome for the different tasks.

Fig. 1. Visual programming platform prototyping

For the mobile reporting part of the PELARS system we adopted a web-based system developed by colleagues [9] that allowed us to create a series of forms that could be accessed by students in an Interaction Design course where they have a 4 week block in physical computing. The students needed to fill in three forms, the first form asks them to briefly describe their plan for solving the task, the second form allows them to document their progress with text and photos, and the final form asks them to reflect on the outcome, did the project succeed as planned. Figure 2 shows the different screens of the mobile system. The intention of our prototyping effort has been to explore the similarities between practice-, problem-, and inquiry-based learning [3] and the challenges in student self-documentation practices in physical computing.

In addition to the forms, the students were also asked to complete a lightweight pre-survey and post-survey to evaluate the usability of the mobile documentation tool. The surveys were inspired by Read and MacFarlane's [6] work on surveys for children in computer interaction and designed to take a few minutes to fill out. The survey results were intended to supplement the submissions received through the mobile system. The pre-survey intended to cover their general experience with documenting their work and the post-survey their views on the usefulness of the tool. Additionally, the pairs of students were interviewed in a semi-structured after the prototyping session.

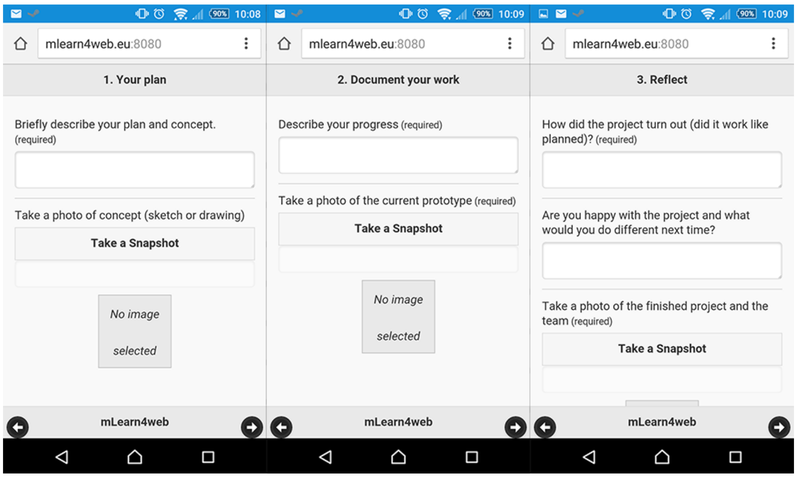

Fig. 2. Mobile system screen captures

3 Initial Results

3.1 Visual Programming

The initial results for the visual programming platform points towards the less experienced programmers finding the visual programming system easier for solving the different tasks. The less experienced students were more open to exploring how to solve the open-ended tasks. While the experienced programmers were frustrated by their perceived limitations of the system, for example not being able to code a loop statement to blink an LED. During the post activity interview, the experienced programmers did however see the system as useful both for learning programming but also for communicating ideas in a prototype stage. Importantly to note, that these perceptions may reflect that design students are more used to open-ended tasks and familiar with throw-away prototyping.

In some cases, the designers worked with more experienced programmers and in these cases communication between the team members helped the programmer shift metaphors to a more visual style of programming. After the initial tasks the more experienced programmers felt they had a better understanding of the concept. Additionally, in the follow-up interviews, they expressed that they liked the idea of visual coding, but primarily saw it as a teaching tool or a communication tool rather than something that they would use to build their projects.

3.2 Documentation Tools

The mobile tools initially seemed to have the right balance of short text entries and the uploading of rich media. The aim was to allow the students to plan easily, document and reflect via smartphones or laptops. Our initial findings suggest that the structure of planning, documenting the process and then reflecting on the project was utilised by the students. The students reported in the post-survey that it is easy to forget to document, to ignore it, or do it later. While the submitted documentation captured the students progress, it was also often submitted the day after or when they were finishing their work, rather than at the end of each session. Our thoughts for these results are that students faced the combination of not seeing the relevance of documenting the projects was important and not having practiced documenting their work.

Students reported in the post survey that the usability of the system needs to be improved. For example, they pointed out that the system did not let them go back to add, or amend their documentation. The need for better clarity what happens with the data after they submit it could help with the students. Connecting the documenting tools to their normal work practices and digital tools, like blogs or online portfolios need to be explored. Additionally while documenting some students appeared were frustrated when submitting as a group. The data shows that when students used a personal device they choose to submit individually. This suggests that the group submissions are useful, the students desire to submit individual reports as well.

4 Discussion

We feel that that the low-fidelity and sketching the technology for the PELARS project are important means to design better intelligent support while engaging with the needs of the different users. The PELARS project has been influenced by inquiry-science learning. However, the nature of making and solving problems with physical computing in interaction design courses can be more dynamic and open-ended than more traditional classwork. Prototyping both the programming interface and the documentation tool as parts of the same project, rather than as separate entities gives a broader design approach. This allows us to explore different aspects of the learning environment and test out ideas in pseudo-real world situations. One of the design goals is to support the visual programming activities with intelligent tutoring and means for teachers and students to analyse of time how they programmed and built the different projects. Additionally, the documentation tool provides a different perspective to the ambient data collection and a process framework for the learning activity. We feel that using these different design approaches provides us with a means to explore the complexity of project-based experiential learning scenarios.

Acknowledgments. The Project is a European Union Research Project for Small or Medium-scale Focused Research Project (STREP) part of the FP7-

ICT-2013-11 Objective ICT-2013.8.2 Technology-enhanced Learning. The Grant Agreement number is 619738. http://www.pelars-project.eu/.

References

1. Blikstein, P.: Multimodal learning analytics. In: Proceedings of the third international conference on learning analytics and knowledge. pp. 102–106. ACM (2013)
2. Buxton, B.: Sketching user experiences: getting the design right and the right design (interactive technologies). Burlington, MA: Morgan Kaufmann (2007)
3. Hmelo-Silver, C.E.: Problem-based learning: What and how do students learn? Educational psychology review 16(3), 235–266 (2004)
4. Lai, E.R., Viering, M.: Assessing 21st century skills: Integrating research findings. In: annual meeting of the National Council on Measurement in Education, Vancouver, BC, Canada (2012)
5. Mavrikis, M., Dragon, T., McLaren, B.M.: The design of wizard-of-oz studies to support students learning to learn together. In: Proceedings of the International Workshop on Intelligent Support in Exploratory Environments: Exploring, Collaborating, and Learning Together (2012)
6. Read, J.C., MacFarlane, S.: Using the fun toolkit and other survey methods to gather opinions in child computer interaction. In: Proceedings of the 2006 conference on Interaction design and children. pp. 81–88. ACM (2006)
7. Slotta, J.D., Tissenbaum, M., Lui, M.: Orchestrating of complex inquiry: Three roles for learning analytics in a smart classroom infrastructure. In: Proceedings of the Third International Conference on Learning Analytics and Knowledge. pp. 270–274. ACM (2013)
8. Suthers, D.D., Dwyer, N., Medina, R., Vatrapu, R.: A framework for conceptualizing, representing, and analyzing distributed interaction. International Journal of Computer-Supported Collaborative Learning 5(1), 5–42 (2010)
9. Zbick, J., Jansen, M., Milrad, M.: Towards a web-based framework to support end-user programming of mobile learning activities. In: Advanced Learning Technologies (ICALT), 2014 IEEE 14th International Conference on. pp. 204–208. IEEE (2014)