Workshop on Developing a Generalized Intelligent Framework for Tutoring (GIFT): Informing Design through a Community of Practice

held in conjunction with

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

Friday, June 26, 2015
Madrid, Spain

Workshop Co-Chairs:

Benjamin Goldberg[1], Robert Sottilare[1], Anne Sinatra[1],
Keith Brawner[1], Scott Ososky[1,2]

[1] *U.S. Army Research Laboratory, Orlando, FL 32826*
[2] *Oak Ridge Associated Universities, Oak Ridge, TN 37830*

https://gifttutoring.org/news/42

Table of Contents

Preface

The purpose of this workshop is to examine current research within the AIED community focused on improving adaptive tools and methods for authoring, automated instruction and evaluation associated with the Generalized Intelligent Framework for Tutoring (GIFT). As GIFT is an open-source architecture used to build and deliver adaptive functions in computer-based learning environments (Sottilare, Brawner, Goldberg & Holden, 2013), this workshop aids in gathering feature requirements from the field and addressing issues to better support future users.

The topics of interest highlight current research conducted within the GIFT community (i.e., 400+ users in 30+ countries) across three themes: (1) modeling across affect, metacogntion, teams, and experts; (2) tutorial intervention through communication, guidance, and sequencing; and (3) persistence functions of intelligent tutoring associated with competency modeling and social media. Each theme will be comprised of short papers describing capability enhancements to the GIFT architecture, the motivation behind the described work, and considerations associated with its implementation. Paper presentations are organized to provide attendees with an interactive experience through hands-on demonstrations.

For attendees unfamiliar with GIFT and its project goals, this workshop exposes those individuals to the GIFT architectural structure, enabling participants to learn how to construct original functions, and how the framework can be applied to their own research. The intent is to engage the AIED community in an in-depth exploration of the various research topics being investigated and the potential leveraging and collaboration that a community framework such as GIFT affords.

Benjamin Goldberg, Robert Sottilare, Anne Sinatra, Keith Brawner, Scott Ososky
The GIFT 2015 Co-Chairs

References

1. Sottilare, R., Brawner, K. W., Goldberg, B., & Holden, H. (2013). The Generalized Intelligent Framework for Tutoring (GIFT). In C. Best, G. Galanis, J. Kerry & R. Sottilare (Eds.), *Fundamental Issues in Defense Training and Simulation* (pp. 223-234). Burlington, VT: Ashgate Publishing Company.

Challenges in Moving Adaptive Training & Education from State-of-Art to State-of-Practice

Robert A. Sottilare[1]

[1]U.S. Army Research Laboratory, Orlando, FL 32826
Robert.a.sottilare.civ@mail.mil

Abstract. Adaptive training and education (ATE) systems are the convergence of intelligent tutoring system (ITS) technologies and external training and education capabilities (e.g., serious games, virtual humans and simulations). Like ITSs, ATEs provide instructional experiences that are tailored to the learner and may be more effective than the training or educational systems alone. ATEs also leverage existing environments, content and domain knowledge to reduce the authoring workload. The Generalized Intelligent Framework for Tutoring (GIFT) is an open-source ATE architecture with the primary goal to support easy authoring, automated instructional management during ATE experiences, and a testbed to evaluate the effect of ATE tools and methods. While this paper addresses challenges and goals in bringing ATE solutions from state-of-art to state-of-practice within GIFT, it also highlights generalized challenges in making ITS technologies ubiquitous and practical on a large scale across a broader variety of domains.

Keywords: adaptive training and education (ATE), intelligent tutoring system (ITS), authoring, instructional management, domain modeling

1 Introduction

An adaptive training and education (ATE) system is the convergence of Intelligent Tutoring Systems (ITS) technologies and what might normally be standalone training and educational capabilities (e.g., serious games, virtual humans, and virtual, mixed, and augmented-reality simulations). The resulting integration provides intelligently-tailored, computer-guided learning experiences for both individual learners and teams which leverages and enhances the capabilities of existing training and educational infrastructure.

ATE research is focused on optimizing performance, efficiency (e.g., reduced time to competency) deep learning (e.g. higher retention and reduced need for refresher training), and transfer of skills to the operational environment (on the job). The Generalized Intelligent Framework for Tutoring (GIFT) is an open-source, modular architecture whose goals include reducing the cost and skill for authoring ATE systems, automating instructional management, and tools for the evaluation of ATE technologies [1]. GIFT was created to capture best instructional practices and the results of enabling ATE research objectives including ITS design, data analytics, human-system interaction, automated authoring, and the application of learning theory.

Several ATE integration tools and prototypes have been created and are being evaluated. The Game-based Architecture for Mentor-Enhanced Training Environments (GAMETE), is a middleware tool to integrate serious games (e.g., Virtual Medic) and tutors (e.g., GIFT-based tutors and AutoTutor Lite tutors) [2]. The Student Information Model for Intelligent Learning Environments (SIMILE) is a tool for linking actions in games to ITS learning measures [3]. Newtonian Talk is the integration of Physics Playground, AutoTutor, and GIFT [4] to support interactive discovery learning of key physics principles. Virtual Battle Space 2, a serious military training game, has also been integrated with GIFT [5]. As a result of developing and evaluating these prototype ATE tools and systems, lessons-learned and several challenge areas have been identified.

2 Challenges, Goals, and Objectives

The idea of generalizing the authoring of ITSs for broad application across task domains (cognitive, affective, psychomotor, and social) ranging from simple to complex, and from well-defined to ill-defined is not a new goal [6, 7]. However, there remain several challenges in realizing a generalized tutoring architecture to produce standalone ITSs and integrated ATE systems. We have identified seven challenge areas or barriers to adoption of ATE technologies: affordability and efficiency; adaptability and persistence; accuracy and validity; relevance and generalizability; accessibility; credibility; and effectiveness.

Each of these challenges could also be considered a desired characteristic or end state. While all of the seven challenges may be considered on the critical path to practical ATE systems, the challenges which impact authoring and learner modeling are most critical. The authoring process is critical to affordability and is therefore the most significant barrier to adoption.

Accurate learner modeling is critical to the whole instructional decision process for ATE systems. To fully understand the learner's states and adapt instruction to optimize learning and mitigate barriers to learning, ATE systems (and ITSs) need to meet two challenges: low cost, unobtrusive methods to acquire learner behavioral and physiological data; and highly accurate, near real-time classification methods for learner states based on behavioral and physiological data. The effect of adaptive instruction on learner states and specifically critical learning moderators [8] (e.g., engagement, motivation) is illustrated in Figure 1.

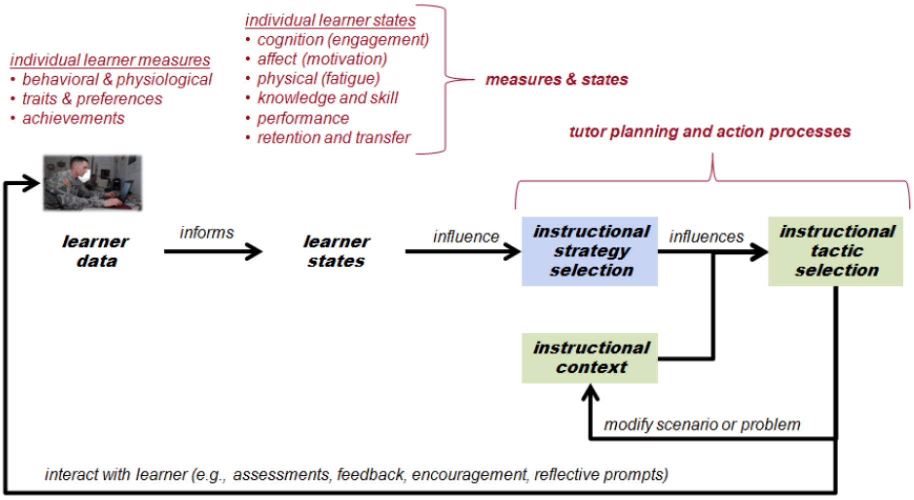

Fig. 1. Updated Learning Effect Model

Inaccurate modeling of learner states can lead to the selection of less than optimal strategies and tactics. Negative outcomes include the selection of instructional strategies which either confuse or frustrate the learner to the point of withdrawal or provide negative training effects because the strategy selected is in opposition to the learner's actual state.

The following is a discussion of the seven challenges and their associated goals and objectives along with a projected impact on adoption in the context of associated ATE/ITS processes: authoring, maintenance, individual learner and team modeling, instructional management, domain modeling, user interface design, and architecture.

2.1 Challenge: Affordable, Efficient, and Effective Adaptive Systems

Due to high authoring costs, the investment in ITS development is only practical for high density courses, those with a high student population. ITS and ATE system developers be able to define what a pound of adaptive training and education is worth in comparison to alternative methods, and they must be able to quantify a return-on-investment and associated breakeven points for these investments [9]. Adaptive systems by their nature require the authoring of additional content and domain knowledge.

To make ATE technologies affordable, we must first examine the authoring and maintenance processes. Aleven, McLaren, Sewall and Koedinger [10] assert that it takes approximately 200-300 hours of development time to author one hour of adaptive instruction. This assertion is based on well-defined, cognitive (e.g., problem solving and decision-making) domains. Research is needed to define the authoring time for more complex, ill-defined domains. A goal for GIFT designers is to reduce authoring time for any domain to just a few hours. This would make it practical for teachers, course managers, and other domain experts to rapidly develop adaptive content and make courses agile and adaptive to learner needs.

However, in the case of ATE systems, we are looking at a broader definition of domain complexity with ill-defined domains and non-cognitive tasks and factors. So given we are developing more complex instruction, our goal is not just to reduce the time and cost to author ATE systems, but also to reduce the skills required to develop and maintain standalone ITSs and integrated ATE systems. To meet this goal, we must improve interoperability and reuse of ITS components and domain knowledge, automate authoring processes wherever possible to take humans out of the loop, improve curation (search, retrieval, management) of domain knowledge, and improve user interfaces to enhance authoring efficiency (ease of use) where human-in-the-loop authoring is required.

What will it take to make ATE authoring available to the masses? A goal is for domain experts to be able to author ATE systems without knowledge of computer programming, instructional design principles, or learning theory. These would be integral to ATE design along with automated authoring tools and artificially intelligent job aids which will guide authors efficiently through the end-to-end development process in the future. As part of the authoring process, we advocate standards to make integration of external training and education systems with ITS easier. Fixing the authoring process is a “must do” to make ATE systems practical (affordable, effi-cient, and effective).

2.2 Challenge: Enhance Adaptability and Persistence

The adaptability of ATE systems is limited when compared to expert human tutors. Our goal is to enhance the ability of ATE systems to provide unique learning experiences for each and every learner. ATE systems by their nature require additional content and associated domain knowledge to support a broad population of learners. This fact drives the cost of ATE systems and limits options for tailoring of ATE experiences for individual learners and teams of learners. By finding tools and methods to reduce the time/cost and skills required to author ATE systems, we can provide more tailoring options in the same or less development time.

Another area for improvement in ATE systems design is in individual learner and team modeling. Our objectives are to enhance short-term and long-term learner modeling to improve the adaptability of ATE systems. Research is needed to understand the relationship between desired outcomes (e.g., learning, performance, retention, and transfer) and the learner’s behaviors, transient states (e.g., goals, affect), trends and cumulative states (e.g., domain competency and prior knowledge), and their enduring traits (e.g., personality, gender, and first language) in order to facilitate efficient learner modeling, optimized instructional decisions, and thereby authoring of ATE systems. Adaptive instruction based on long-term modeling of the learner will offer persistence not present in today’s ITSs. We can enhance adaptability by making learner and team modeling central to instructional decisions made by ATE systems.

2.3 Challenge: Enhance Accuracy and Validity of Instructional Decisions

In order to make appropriate adaptive instructional decisions, ATE systems need to improve their perception of learner states. Research is needed to develop low cost, unobtrusive methods to acquire learner data to support state classification. In turn, research is also needed to improve the accuracy of real-time classification for both individuals and units [11].

To insure the validity (suitability) of instructional decisions based on sound learning theory, domain-independent instructional strategies (e.g., metacognitive prompts) may be selected based on the accurate classification of the learner's states. For example, imagine a learner whose state is classified as "confusion" by an ATE. If the accuracy of this classification is less than 80 percent, then a metacognitive prompt to have the learner reflect on a recent decision could clarify any ambiguity of the "confusion" classification.

Similarly, domain-specific actions (tactics) based on a selected instructional strategy and context (conditions within the domain). Research is needed to develop methods to optimally select the best possible strategies and tactics given the learners states, the conditions within the training or educational domain, and the availability of options provided by the author of the ATE. Within GIFT, the learning effect model for individual learners [11, 12, 13], as updated in Figure 1, describes the interaction between the learner and the ITS.

2.4 Challenge: Enhance Task Relevance & Implement Generalized Solutions

In order to be practical, ATE systems must be able to represent domain knowledge in relevant task domains. Today, the most popular ITS domains are mathematics, physics, and computer programming. The characteristics of other domains may not be as well defined or as simple. For example, tasks involving psychomotor and perceptual measures (e.g., sports, laparoscopic surgery, and marksmanship) are not well-represented in the ITS community.

Research is needed to expand the dimensions of domain knowledge to support a broader variety of task domains. One objective is to develop standards to represent domain knowledge beyond the cognitive task domain (e.g., affective, psychomotor, perceptual, social, ill-defined, and complex domains). Once the domain can be represented, authoring tools and instructional strategies, tactics, and policies should be tailored to support adaptive interaction with the learner.

As mentioned previously, it will be critical to be able to easily integrate external training and educational environments to reduce the authoring burden, but also to enhance the experiences that are familiar to learners. Representing the domain knowledge of relevant task domains and integrating with other systems will provide the basis for an ATE architecture which we are currently prototyping as GIFT.

2.5 Challenge: Support Tutoring at the Point-of-Need

To be effective, ATE must be accessible at the user's point-of-need. The ATE architecture must develop services to allow access anyplace and anytime (24/7/365). To meet this goal we have formulated two primary objectives. The first is to move GIFT, an adaptive training and education architecture, to the cloud. We are developing a cloud-based architecture that allows real-time access for learners and units to support individual, collaborative (social), and team training and education. Since learners, authors, and other ATE system users may find themselves in areas of degraded communications, we are also developing cloud-based services to download virtual machine versions of GIFT to allow local development and synch with the cloud as needed.

2.6 Challenge: Enhance the Credibility and Supportiveness of the Tutor

To enhance the learner's perception of ATEs as credible training and educational tools (e.g., domain experts, trusted advisors, teachers), we are closely emulating best practices of expert tutors and learning theory. To this end we have implemented component display theory [14] as our default pedagogical module, the engine for managing adaptive pedagogy or eMAP.

To capture and maintain the attention of learners, we are developing methods to evaluate the suitability of user interfaces (e.g., virtual humans) and domain knowledge (e.g., content) to enhance the learner's perception of ATE systems with respect to domain expertise and learner support. To be efficient, we are developing user inter-faces for various roles in the ATE environment (e.g., learners, authors, and power-users). These interfaces will allow users to construct their own mental models and interact in a manner that is conducive to learning.

2.7 Challenge: Continuously Evaluate Effectiveness

As with many systems, we anticipate that ATE systems will be deployed with implementations of *best known practices*. ATE systems must not only provide adaptive instruction, but be adaptive to continuously improve. The challenge is to collect and analyze large datasets on a regular basis to identify trends and issues, and to evaluate the effectiveness of current tools and methods against alternative tools and methods. The ATE architecture must be able to support continuous evaluation of its tools and methods, and be modular in order to support rapid change.

We are developing tools and methods within GIFT to evaluate the effectiveness of the authoring and instructional management processes. Our goal is to support the continuous improvement of ATE technologies. To this end we are developing tools and methods to reduce the time/cost and skill required to evaluate the effectiveness of ITS technologies. We are also developing data analytic methods to evaluate user-generated content (social media) to maintain cognizance of the primary users (learners and authors) and to enable them as change agents.

3 Conclusions

This paper reviewed several challenges to adoption of ATE systems as practical tools for learning. We noted that several ongoing research initiatives and identified several more which are needed to support changes to the authoring and maintenance, instructional management, learner modeling, and domain modeling processes along with underlying services provided by the architecture through the user interface.

We also noted that ATE systems have a long-term focus as well as a short-term learning focus. Big data collected continuously on both the learner populations and the ATE system may be analyzed to provide insight on both effective and ineffective instructional methods and user interfaces for both authoring and instruction. Research is still needed to fully understand the effect of combining ITSs with existing training and education systems in order to quantify a return-on-investment.

We recommend additional research emphasis on the following challenge problems: methods to automate the authoring process to the maximum extent possible; enhanced job aids and user interfaces for the authoring process where automation is not possible yet; methods to automate integration of existing training and education systems with ITSs; methods to increase the accuracy of learner state classification and optimize instructional decisions by the tutor; methods to evaluate the effectiveness of ATE system tools and methods; and methods to evaluate user-generated content (e.g., social media) to enhance learner experiences in ATE systems.

We also note the need to expand ITSs beyond the existing well-defined domains (e.g., mathematics, physics, and computer programming) to include more ill-defined and dynamic domains (e.g. psychomotor domains including sports). Finally, we advocate the development of collective level models (e.g., shared states, team behaviors, and team cohesion) for unit-level tasks and collective learning environments [15].

References

1. R. Sottilare, K. Brawner, B. Goldberg & H. Holden. The Generalized Intelligent Framework for Tutoring (GIFT). Orlando, FL: U.S. Army Research Laboratory – Human Research & Engineering Directorate (ARL-HRED), 2012.
2. J. Engimann, T. Santarelli, W. Zachary, X. Hu, Z. Cai, H. Mall & B. Goldberg, "Game-based Architecture for Mentor-Enhanced Training Environments (GAMETE)", In R. Sottilare (Ed.) *2nd Annual GIFT Users Symposium* (GIFTSym2), Pittsburgh, Pennsylvania, 12-13 June 2014. Army Research Laboratory, Orlando, Florida. ISBN: 978-0-9893923-4-1
3. H. Mall & B. Goldberg, "SIMILE: An Authoring and Reasoning System for GIFT", In R. Sottilare (Ed.) *2nd Annual GIFT Users Symposium* (GIFTSym2), Pittsburgh, Pennsylvania, 12-13 June 2014. Army Research Laboratory, Orlando, Florida. ISBN: 978-0-9893923-4-1
4. M. Ventura, X. Hu, B. Nye & W. Zhao, "NewtonianTalk: Integration of Physics Playground and AutoTutor using GIFT", In Proceedings of the "Developing a Generalized Intelligent Framework for Tutoring (GIFT): Informing Design through a Community of Practice" Workshop at the *17th International Conference on Artificial Intelligence in Education* (AIED 2015), Madrid, Spain, June 2015.

5. B. Goldberg, R. Sottilare, K. Brawner & H. Holden, "Adaptive Game-Based Tutoring: Mechanisms for Real-Time Feedback and Adaptation", *International Defense & Homeland Security Simulation Workshop in Proceedings of the I3M Conference*. Vienna, Austria, September 2012.
6. T. Murray. Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education*, 1999, 10(1):98–129.
7. T. Murray. "An Overview of Intelligent Tutoring System Authoring Tools: Updated analysis of the state of the art", *Authoring tools for advanced technology learning environments*. 2003, 491-545.
8. M. R. Lepper, M. Drake, & T. M. O'Donnell-Johnson. Scaffolding techniques of expert human tutors. In K. Hogan & M. Pressley (Eds), Scaffolding student learning: Instructional approaches and issues. New York: Brookline Books, 1997, 108-144.
9. J. Fletcher & R. Sottilare. Cost Analysis for Training & Educational Systems. In R. Sottilare, A. Graesser, X. Hu, & B. Goldberg (Eds.) Design Recommendations for Intelli-gent Tutoring Systems: Volume 2 - Instructional Management. Army Research Laborato-ry, Orlando, Florida, 2014. ISBN: 978-0-9893923-2-7.
10. V. Aleven, B. McLaren, J. Sewall & K. Koedinger, "The Cognitive Tutor Authoring Tools (CTAT): Preliminary Evaluation of Efficiency Gains", Proceedings of the *8th International Conference on Intelligent Tutoring Systems*, 2006, 61-70.
11. R. Sottilare, C. Ragusa, M. Hoffman & B. Goldberg, "Characterizing an adaptive tutoring learning effect chain for individual and team tutoring". In Proceedings of the *Interservice/Industry Training Simulation & Education Conference*, Orlando, Florida, De-cember 2013.
12. J.D. Fletcher & R. Sottilare. Shared Mental Models of Cognition for Intelligent Tutoring of Teams. In R. Sottilare, A. Graesser, X. Hu, & H. Holden (Eds.) *Design Recommendations for Intelligent Tutoring Systems: Volume 1- Learner Modeling*. Army Research Laboratory, Orlando, Florida, 2013. ISBN 978-0-9893923-0-3.
13. R. Sottilare. Considerations in the Development of an Ontology for a Generalized Intelligent Framework for Tutoring. *International Defense & Homeland Security Simulation Workshop* in Proceedings of the *I3M Conference*. Vienna, Austria, September 2012.
14. D. Merrill, B. Reiser, M. Ranney, & J. Trafton, "Effective Tutoring Techniques: A Comparison of Human Tutors and Intelligent Tutoring Systems". *The Journal of the Learning Sciences,* 1992, 2(3), 277-305.
15. R. Sottilare. Special Report: Adaptive Intelligent Tutoring System (ITS) Research in Support of the Army Learning Model - Research Outline. *Army Research Laboratory* (ARL-SR-0284), December 2013.

Learning Ecosystems Using the Generalized Intelligent Framework for Tutoring (GIFT) and the Experience API (xAPI)

Michael Hruska[1], Ashley Medford[1], Jennifer Murphy[2]

[1]Problem Solutions 1407 Eisenhower Blvd., Johnstown, PA 15904
[2]Quantum Improvements Consulting 812 Oak Chase Dr., Orlando, FL 32828

Abstract. Learning ecosystems provide a combination of technologies and support resources available to help individuals learn within an environment [1]. The Experience API (xAPI) is an enabling specification for learning ecosystems, which provides a method for producing interoperable data that can be shared within a learning ecosystem [2]. Version 4.1 of the Generalized Intelligent Framework for Tutoring (GIFT) provides support to both produce and consume xAPI data. A number of use cases are enabled by this support. This paper will explore the use cases, functionality enabled, setup and design guidance in addition to exploring practical applications for using GIFT and xAPI within learning ecosystems.

Keywords: adaptation, Experience API, intelligent tutoring systems, learning, xAPI, GIFT, computer-based tutoring systems, learning ecosystems

1 Introduction

Organizations in the U.S. alone invested approximately $164.2 billion on employee training and development in 2012 [3], and in 2013, an average of over $1,200 per employee was spent for direct learning [4]. With 38% of this training being delivered using technology [4], this investment is increasingly being spent on non-traditional training methods and technologies. As learning ecosystems continue to grow in complexity, so too do the challenges faced by education and training professionals.

Personalizing education and assessing student learning are grand, educational challenges being faced today [5]. Recent efforts on learning ecosystems reflect this movement towards adaptive and tailored learning [5,6]. In general, the goal in a learning ecosystem is to leverage performance data in order to assess and adapt learning and in turn, increase training effectiveness and lower associated training time and costs [6]. By capturing the massive amount of learning data tied to each individual and bound within a learning ecosystem, the ability to meet these educational challenges by intelligently tailoring learning and assessing performance is possible.

Research and development efforts by the Advanced Distributed Learning (ADL) initiative of the Department of Defense (DoD) and the U.S. Army's Research Laboratory (ARL) are striving to meet these complex challenges. The Experience API specification (xAPI), developed by ADL, provides an interoperable means to describe and track learning in various learning ecosystem components [7]. ARL's work on interoperability of performance data and intelligent tutors, specifically the Generalized Intelligent Framework for Tutoring (GIFT), along with xAPI provide a basis for this paper. The use of xAPI in conjunction with intelligent tutoring (e.g., GIFT) permits the creation of a reference architecture and provides functionality for a number of use cases. Installation and configuration of open source software components enable testing and experimentation around these use cases. This paper outlines the technical information, reference architecture, use cases, configuration, and expected behaviors of the technology components surrounding this work.

1.1 Existing Efforts

The ARL effort on Interoperable Performance Assessment (IPA) focuses on uniformly defining and describing learning experiences [8]. IPA defines methods for encoding human performance data using xAPI statements [9]. The goal of such encoding is to create data with *inter-system data value* to support adaptation in learning ecosystems. Additionally, interoperable encoding can provide rich data analytics and visualizations.

ARL's IPA research works primarily toward the goal of defining uniform performance measures in simulation and providing summative assessments towards these measures from multiple sources. Additional IPA efforts, focused on using small group and team data, also indicate the potential of such approaches to adapt and even drive team formation [10]. Overall, IPA efforts aim to address the following use cases: show a historical view of proficiency; show a live view of performance; enable macro and micro training adaptation, and; collect Big Data for trends analysis.

1.2 Experience API and Learning Ecosystems

The xAPI is a supporting specification for learning ecosystems. The xAPI specification defines an interface for a common and interoperable data store for xAPI statements, known as a Learning Record Store (LRS). The LRS provides a single storage point in a learning ecosystem. Systems within a learning ecosystem either act as a "producer" of xAPI statements or as a "consumer". [7]

1.3 The Generalized Intelligent Frameworks for Tutoring (GIFT)

GIFT, developed by ARL's Human Research and Engineering Directorate (HRED), provides a service-oriented framework of tools, methods and standards to make it easier to author computer-based tutoring systems (CBTS), manage instruction, and assess the effect of CBTS, components and methodologies [11]. GIFT was enhanced to interoperate with xAPI in Version 3.02 to provide a *consumer* functionality and in version 4.1 to provide *producer* functionality.

1.4 Reference Architecture

The Figure below (Fig. 1) shows a reference architecture for a learning ecosystem using GIFT.

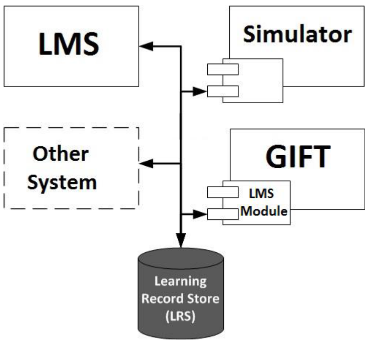

Fig. 1. A reference architecture for a learning ecosystem is shown [12]. The architecture shows a *Learning Record Store (LRS)* where data is stored and retrieved by elements of the ecosystem. A simulator or other system(s) may produce or consume data that is stored in the LRS. *GIFT* uses the *LMS Module*, which is enabled to both produce and consume xAPI data via the LRS submodule. GIFT is thus able to provide interoperability between these other systems using their xAPI data.

The architecture is composed of components that might comprise the learning ecosystem like a Learning Management System (LMS), a Simulator, GIFT, and other systems such as games or virtual worlds. In the example, the use of xAPI data as a common data format enables the LMS, GIFT, and other systems to be interoperable. The xAPI data created by the systems is stored in the LRS. In turn, xAPI data pulled from the LRS may be consumed by any of the systems within the ecosystem. Notably, GIFT provides both consumer and producer functionality as it (a) produces xAPI statements for other elements in the ecosystem and (b) consumes xAPI statements [12].

1.5 Use Cases

A number of use cases for learning ecosystems are supported by GIFT and its xAPI functionality. GIFT may be used in conjunction with an LRS and other systems to demonstrate and test these use cases. The following are some potential use cases that may be built upon GIFT and the xAPI functionality:

1. **Multiple System Performance Assessment.** Multiple systems including live scenarios using observer based tools, simulations, LMS, and games can be utilized to assess performance and produce xAPI data. Multiple systems can be used to assess

a singular competency or set of competencies across multiple delivery modalities to demonstrate performance over time. This data can be employed to drive adaptation as GIFT acts as a consumer.

2. **Using Simulation for Assessment.** A simulation may be used for performance assessment. The simulation produces xAPI data. This data may also be used to drive adaptation as GIFT acts as a consumer.
3. **GIFT-Driven Data Production.** xAPI data about course content and concepts contained within a course can be created and stored in an LRS. This data provides granular evidence of a user's interaction with a course and its corresponding concepts.
4. **Macro-Adaptation.** GIFT can provide macro adaptation or outer loop adaptation based upon the data it consumes. Performance deficiencies produced by GIFT or other systems that are stored as xAPI data can be used to intelligently navigate or recommend courses or other learning experiences. For example, a learner uses a simulator for marksmanship training and is found deficient in breathing techniques. The next time the learner logs into GIFT, he/she would then receive training recommendations such as courses or additional simulator training to improve their breathing techniques. In other words, GIFT leverages xAPI data about a user's deficiencies that is produced within a single learning event and then provides recommendations or adapts the individual's overall learning path to address these deficiencies.
5. **Inter-System Driven Micro-Adaptation.** GIFT can provide micro-adaptation within a scenario based upon data it consumes from other systems. For example, a learner participates in several marksmanship simulations and is found deficient in breathing techniques. Leveraging this xAPI data from one or multiple learning events, a future marksmanship simulator adapts within its scenario by providing additional guidance for breathing techniques. In other words, GIFT is able to leverage past xAPI data produced by other systems to drive micro-adaptation within future learning events in other systems.

2 Using GIFT and xAPI

GIFT (Version 4.1) is capable of both producing and consuming xAPI statements. Minimal configuration is required to setup this functionality in GIFT. Version 4.1 natively supports use cases 1, 2, 3, and 4 outlined in Section 1.5. Additional programming related to content development is required to support use case 5.

2.1 GIFT LMS/LRS Module

The LMS module within GIFT, responsible for retrieving and storing training and assessment history, enables xAPI support. The LMS module has been enhanced by creating an LRS submodule within which it allows both polling of and writing to the LRS.

2.2 Setting up GIFT with xAPI Support

In order to enable xAPI functionality for GIFT, an LRS must be available and connected to the network which GIFT is installed on. The following steps need to be completed to enable xAPI support in GIFT:

1. Install GIFT framework (refer to www.gifttutoring.org)
2. Install an LRS (see below)
3. Configure GIFT to communicate with the LRS end point

Several open source LRS options exist as well as commercial options. The following open source LRS solutions are currently available:

- Open source LRS from ADL - https://github.com/adlnet/ADL_LRS
- Hosted LRS from ADL- https://lrs.adlnet.gov/xapi/
- Open source LRS from learning locker - http://learninglocker.net/

Configuration of xAPI End Point. Once GIFT and the LRS are installed, GIFT must be configured to communicate with the LRS endpoint. The following steps must be undertaken to allow GIFT to communicate with the LRS:

1. Open the LMSConnections.xml file located in the `<GIFT Root>\GIFT\config\lms` directory
2. Select edit, and add a new connection entry under the `<LMSConnections>` root using the following information format and entering the `username`, `password`, and `URL` for the LRS installation between the XML elements:

```
<Connection>
  <enabled>true</enabled>
  <impl>lms.impl.Lrs</impl>
  <name>LRS Name</name>
  <Parameters>
    <networkAddress>https://lrs.url</networkAddress>
    <username>username</username>
    <password>password</password>
</Parameters>
</Connection>
```

2.3 GIFT as a Producer of Interoperable Data

Once configured, GIFT is enabled to act as a producer of xAPI data. As a producer, once a training scenario is completed, the course records and scores are passed to the LMS module for storage. This data is then passed to the LMS database as well as the LRS sub-module. An xAPI statement is generated for each level of the graded score nodes, and each statement is linked to their parent statement. The figure (Fig. 2.) below outlines an example of data that is created and defined for the elements in the xAPI format.

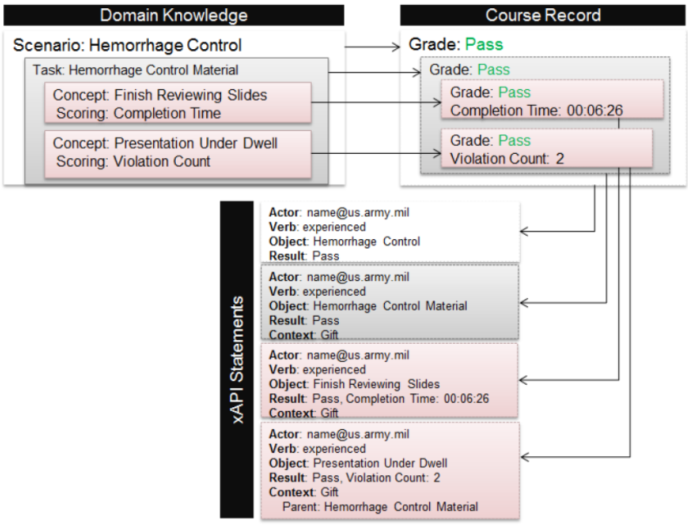

Fig. 2. An example of data from a *Domain Knowledge File, Course Record,* and *xAPI Statements* is shown. The example outlines the scenario, tasks, concept, and grades that are used to define the xAPI data elements. [12]

Editing Domain Knowledge File. In order for GIFT to produce xAPI data, the concepts that are represented within a course must be added to the XML file that represents the course. The following steps must be taken to update the file:

1. Edit the XML file for the course located at `<GIFT Root>\Domain`
2. Add a `<concepts>` section under the `<Course>` root. Below is an example of the addition of the `<concepts>` elements:

```
<Course name="Course Example"...>
   ...<concepts>
      <concept>Skill 1</concept>
      <concept>Skill 2</concept>
      <concept>Skill 3</concept>
   </concepts>...
</Course>
```

2.4 GIFT as a Consumer of Interoperable Data

The LMS module of GIFT also provides consumer functionality. The consumer function allows GIFT, via the LRS submodule, to poll the LRS end point. xAPI statements are used to extend GIFT's course suggestion capabilities. The LMS polling function retrieves a user's history, using their email address as an identifier when the user logs into GIFT. The LMS module examines available course metadata definitions to find courses with concepts that match the user's deficiencies. The LMS module then recommends concepts matching deficiencies noted in xAPI statements for which the user is "below" concept proficiency. Dynamic filtering of course suggestions is presented through the "Recommended Courses" (See Fig 3).

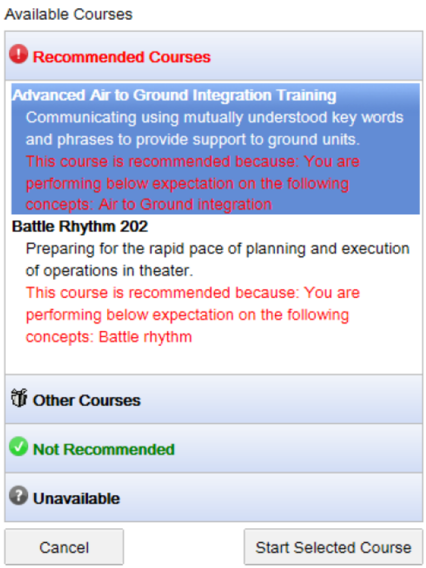

Fig. 3. A screen shot of GIFT *Available Courses* is shown. The example outlines recommended courses as determined by the LMS module by examining course metadata and deficiencies stored in xAPI statements within the LRS. [12]

3 Conclusions

GIFT allows enhanced functionality via its LMS module to integrate external data sources in a learning ecosystem. GIFT also enables data created within GIFT to be stored in an interoperable way that supports learning ecosystems via xAPI in an LRS. This functionality enables GIFT and other systems to evaluate incoming student competencies in order to better inform instructional strategy. Systems in the learning ecosystem are also enabled to make recommendations for the next training events based on performance data.

Using this functionality, researchers may test a number of different use cases and functions of adaptive learning in learning ecosystems. Usage of xAPI data in learning ecosystems with GIFT and other producers will allow consumers in learning ecosystems to assess and tailor learning and ultimately, to leverage Big Data analytics to discover trends over time.

The ability to leverage xAPI data in GIFT enables the investigation of a number of research questions. For example, the Army's current training modernization goals call for the development of persistent representations of Soldier performance in order to support a culture of lifelong learning. In order to develop these complex student models, Soldier performance must be tracked across multiple training environments (e.g., events, simulators, courses). By producing and consuming xAPI statements, GIFT can support interoperable student models. However, while research is ongoing in this area, demonstrating interoperable performance data across multiple platforms through GIFT has yet to be accomplished. Further, the question of how best to remediate student performance using xAPI data through GIFT has yet to be investigated. A major question remains about the specific level of granularity of these xAPI statements that is most appropriate for adapting training through GIFT. It is very likely that as independent researchers develop their own solutions for adapting training based on xAPI data, the level of detail required will depend upon the specific domain and application. For the Army to reach its goal of tracking performance across a Soldier's career, however, there must be some consensus on how to standardize the granularity of xAPI statements. These, and other research questions, provide possibilities for research going forward.

References

1. Kelly, D.: What Is a Learning Ecosystem? The eLearning Guild, http://twist.elearningguild.net/2013/11/what-is-a-learning-ecosystem/ (2013)
2. Advanced Distributed Learning: xAPI Specification. GitHub, https://github.com/adlnet/xAPI-Spec (2015)
3. American Society for Training & Development: 2013 State of the Industry. ASTD DBA Association for Talent Development (ATD), Alexandria (2013)
4. Association for Talent Development: 2014 State of the Industry. ASTD DBA Association for Talent Development (ATD), Alexandria (2014)
5. Woolf, B.P: A Roadmap for Education Technology. National Science Foundation #0637190 (2010)
6. Foreman, S. & Rosenburg, M. J.: Learning and Performance Ecosystems: Strategy, Technology, Impact, and Challenges. Santa Rosa: The eLearning Guild, http://www.elearningguild.com (2014)
7. Advanced Distributed Learning: Training and Learning Architecture (TLA): Experience API (xAPI), http://www.adlnet.gov/tla/experience-api (2014)
8. Poeppelman, T., Ayers, J., Hruska, Long, R., Amburn, C., Bink, M.: Interoperable Performance Assessment using the Experience API. In: The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), vol. 2013. (2013)
9. Hruska, M., et al. (2013)
10. Hruska, M., Long, R., Amburn, C., Kilcullen, T., Poeppelman, T.: Experience API and Team Evaluation: Evolving Interoperable Performance Assessment. In: The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), vol. 2014. (2014)
11. Sottilare, R.A., Brawner, K.W., Goldberg, B.S., & Holden, H.K.: The Generalized Intelligent Framework for Tutoring (GIFT). U.S. Army Research Laboratory – Human Research & Engineering Directorate (ARL-HRED), Orlando (2012)
12. Hruska, M., Long, R., Amburn, C.: Human Performance Interoperability via xAPI: Current Military Outreach Efforts. In: Fall Simulation Interoperability Workshop, 14F-SIW-035, Orlando (2014)

Demonstration: Using GIFT to Support Students' Understanding of the UrbanSim Counter Insurgency Simulation

James. R. Segedy[1], John S. Kinnebrew[1], & Gautam Biswas[1]

[1]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. This paper presents our recent work with the Generalized Intelligent Framework for Tutoring (GIFT) for authoring tutors and training systems in concert with already developed external applications that provide a wide variety of educational experiences. In this paper, we describe our efforts to extend the GIFT system to develop metacognitive tutoring support for UrbanSim, a turn-based simulation environment for learning about counterinsurgency operations. We discuss specific extensions to GIFT as well as the links we have developed between GIFT and UrbanSim to track player activities. Additionally, we discuss a conversational approach that we are designing to interpret players' strategies and provide feedback when they adopt suboptimal approaches for their counter-insurgency operations.

Keywords: GIFT, UrbanSim, Scaffolding, Adaptive Support

1 Introduction

The Generalized Intelligent Framework for Tutoring (GIFT) provides a software platform and authoring system for designing, developing, and implementing online and in-class educational programs [1-2]. An important aspect of GIFT that makes it different from a number of conventional tutoring systems is its emphasis on interoperability across a variety of existing training applications (TAs). The overall goals are to reduce the high design and development costs of building computer-based tutors and to increase the reusability of educational applications while also creating engaging and adaptive learning spaces that students can access as needed.

While this is a significant advantage of GIFT, it introduces challenges in the number of use cases that must be considered in order to fully leverage and develop a general framework that is compatible with different forms of available educational resources. In this paper, we present our work in exploiting the GIFT platform to develop a metacognitive tutoring environment for the UrbanSim TA [3], a counter-insurgency (COIN) command simulation developed by the Institute for Creative Technologies at the University of Southern California. We describe the steps involved in developing generalized connectors that are currently tailored to support communication from UrbanSim to GIFT. Our work illustrates the flexibility of the GIFT platform to accommodate dynamic tracking of student activities in the UrbanSim COIN environment. Our overall goals are to simultaneously model student problem solving performance, behavior, and strategies, so that the developed GIFT tutor will provide dynamic support when students are involved in training episodes. Our experiences in developing GIFT to support cognitive and metacognitive tutoring lead to a set of design recommendations for further increasing the capabilities, adaptability, and flexibility of developing a variety of tutor-supported TAs with GIFT. We hope that our experiences and development efforts will help future GIFT developers working with other TAs.

2 UrbanSim

UrbanSim [3] (Figure 1) is a turn-based simulation environment in which users assume command of a COIN operation in a fictional Middle-Eastern country. Users have access to a wealth of information about the area of operation they have been assigned to. This includes: intelligence reports on key individuals, groups, and structures; information about the stability of each district and region in the area of operation; economic, military, and political ties between local groups in the region; the commanding team's current level of population support; and the team's progress in achieving six primary lines of effort. The actions that users take are scenario-specific, but they generally involve increasing the area's stability by making progress along the different lines of effort: (1) improving civil security; (2) improving governance; (3) improving economic stability; (4) strengthening the host nation's security forces; (5) developing and protecting essential services and infrastructure; and (6) gaining the trust and cooperation of the area's population.

Students conduct their operations by assigning orders to available units under their command (*e.g.*, *E CO b* and *G CO a* in Figure 1). To commit their orders, they press the *COMMIT FRAGOS* (FRAGmentary OrderS) button to complete one turn in the simulation environment. The simulation then executes the user's orders; simultaneously, it has access to a sociocultural model and complementary narrative engine that determine the actions of non-player characters in the game, which also affects the simulation results. For example, a friendly police officer may accidentally be killed during a patrol through a dangerous area. These *significant activities* and *situational reports* are communicated to the user, and the results of all activities may result in net changes to the user's population support and line of effort scores (see bottom right of Figure 1).

UrbanSim provides documentation and tutorials that should help students gain an appreciation for the challenges inherent in managing COIN operations. For example, they should learn the importance of maintaining situational awareness, managing trade-offs, and anticipating 2nd- and 3rd-order effects of their actions, especially as the game evolves [3]. They should also understand that their actions themselves produce intelligence (through their consequences as observed in the simulation environment), and, therefore, the need to continually "*learn and adapt*" in such complex domains where the available information is often overwhelming, but at the same time may be incomplete. In other words, students should realize that their decisions produce intelligence that may be critical for decision making and planning during the next set of turns. Students can learn about the effects of their actions by viewing causal graphs provided by their security officer (S2). Users who adopt strategies to better understand the area of operation and its culture by viewing and interpreting the effects of their actions using these causal graphs should progressively make better decisions in the simulation environment as the COIN scenario evolves.

Fig. 1. UrbanSim

3 Developing an Application to Connecting UrbanSim to GIFT

Connecting a TA to the GIFT environment involves creating an interoperability interface. This interface is responsible for reporting the actions performed in the TA (and the resulting TA state) to GIFT while also handling control messages sent by GIFT to the TA to keep the two systems in alignment. The various components and their interactions necessary for connecting UrbanSim and GIFT are shown in Figure 2. UrbanSim produces log files that include information on the actions taken by actors in UrbanSim (and the effects of those actions). To report this information to GIFT, we have authored a Java application that monitors the log files and transmits the data to the interoperability interface, which passes the information to GIFT in a predefined structured format. GIFT can then use this data to tutor the student through a web-based interface.

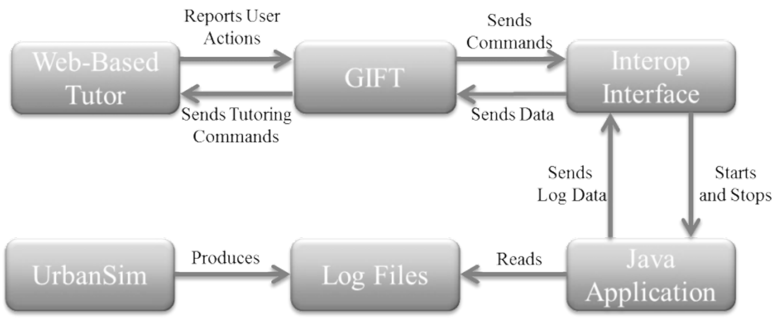

Fig. 2. Communication between GIFT and UrbanSim

The first step in developing this infrastructure required us to create the log parsing application. This involved completing the following steps:

1. Representing the complex set of data models used by UrbanSim.
2. Representing the actions taken by users and the contexts in which they occurred.
3. Monitoring the UrbanSim log directory and translating the log data into the representations created during steps 1 and 2.
4. Implementing code to establish a socket connection with the interop interface and publish the information obtained from the UrbanSim log files.

To represent the data models used by UrbanSim, we reverse engineered the plain-text save files generated by the program, extracted the data objects, their properties, and relationships to other objects and then created 22 Java classes to represent these data models. We then analyzed UrbanSim to extract the set of 38 measurable actions available to students in the program. Finally, we analyzed the set of 19 measurable contexts in which actions could occur. In this instance, a context can be considered to be equivalent to an interface configuration. For example, the configuration shown in Figure 1 shows a map of the area of operation. By tracking the actions and contexts logged by UrbanSim, we were able to create a detailed understanding of students' behaviors in the program. Once these objects had been defined, we focused on developing the algorithm for detecting changes in a log file, extracting the new information, processing it effectively, and then communicating it to the GIFT environment.

Once our log parser application had been written and tested, we turned our attention to writing the GIFT interoperability interface that would connect to the log parser, receive data, and report it to GIFT. To test this functionality, we configured a GIFT performance assessment condition. A condition receives data from the interoperability interface and uses it to assess a student's current level of performance with respect to a concept. In GIFT, a learner model is defined as a set of named concepts that are assessed continually while students are interacting with designated course materials. At any time, each concept may be assessed as being below, at, or above expectation. The data representation is similar to the sampling of a stream: GIFT monitors the student's task performance over time and updates the concept assessments based on the student's most recent performance. Thus, a student may perform above expectation on one concept at some point in the simulated scenario, but fall below expectation on the next turn because they missed a critical piece of information (situational awareness). A history of these assessments is maintained for feedback purposes during a particular learning session and also across multiple sessions. In the tutor we are developing for UrbanSim, the condition we created detects when a student commits their orders and then presents them with a survey through GIFT's tutor user interface, as shown in Figure 3. We expect that the data collected through this survey will provide valuable insight into how students analyze situations in UrbanSim and learn from them as the simulation progresses.

4 Design Recommendations

Our goal in the work is to develop a tutor for UrbanSim using the GIFT framework that can analyze users' understanding of the current COIN scenario, and determine what strategies the user is adopting (if any) in determining their next moves. As we have moved toward this goal, our experiences in coupling UrbanSim and GIFT by authoring a log parsing tool and implementing an interoperability plugin resulted in the following design recommendations to facilitate tutor development:

1. **Expand Instructional Triggers**: GIFT is designed such that all tutoring decisions are bound to changes in a student's concept assessments (below, at, or above expectation). This makes it difficult to author instructional interventions based on non-performance factors. For example, to configure GIFT to show the survey in Figure 3, we had to create a performance assessment condition that detected when the student committed orders and assessed the *committed orders* concept as above expectation (instead of at expectation). The survey was then triggered by a change in the assessment of the committed orders concept. It may be desirable to expand these triggers such that instructional decisions may be directly bound to elapsed time or the occurrence of an event of interest. This could lead to more straightforward authoring of such instruction.
2. **Allow for Contextualized Conversational Instruction and Assessment**: GIFT allows a course author to develop mid-lesson surveys and uses the AutoTutor Lite [4] conversations to administer instructional interventions in appropriate situations. However, the content of these surveys and conversations must be determined ahead of time and may not be parameterized by variables derived from student performance and the state of the system. For example, question 1 in Figure 3 cannot be modified to ask the student about a specific FRAGO that they just committed. Additionally, GIFT does not allow many of these student responses on surveys and in conversations to serve as on-line assessments of their understanding (the exception is that specific answers to multiple choice questions may be linked to assessments of specific concepts). Thus, a student may, in their interactions with surveys and

conversations, reveal information about their understanding that is not utilized in future GIFT interactions. Contextualized conversational feedback has been shown to positively affect learner behavior [5], and so we recommend that such feedback capabilities be incorporated into future versions of GIFT.

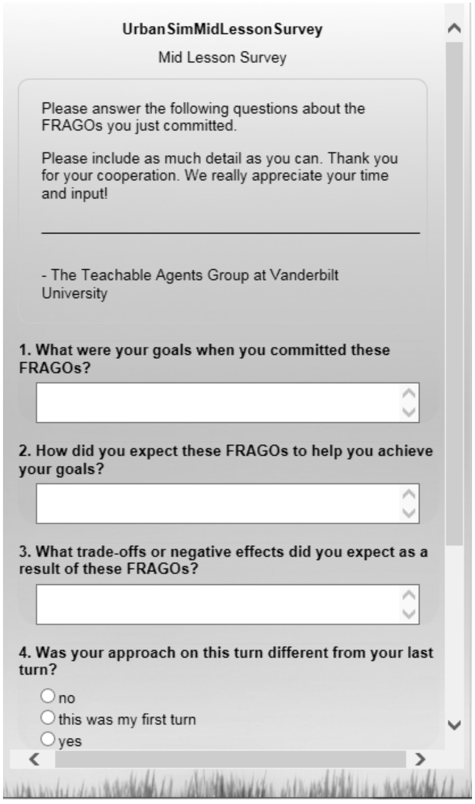

Fig. 3. UrbanSim survey presented through GIFT

3. **Expand Configurability of Dynamic Course Flow**: Currently, the primary structure of a GIFT course is fixed and specified in configuration files. Thus, even if concept assessments show that the student lacks pre-requisite skills, it is difficult to dynamically reconfigure the GIFT course to provide tutorial interventions that help the student develop that skill. In recent versions of GIFT, a system called eMAP [1] has been implemented which allows for dynamic assessment and instruction with regard to mastering a set of domain concepts. While this provides some dynamic capabilities in terms of course flow, we recommend that this system be expanded in the future. In particular, the potential of dynamic GIFT courses could be greatly enhanced with the ability to configure additional aspects of a course or instructional intervention to adapt to the needs of learners. For example, a future version of GIFT could support dynamic flow between *multiple* training applications if

a student's performance in one training application proves that they need training in pre-requisite skills before they are ready to succeed at their task.

5 Conclusions and Future Work

In this paper, we have presented our experiences in creating an application to synchronize the UrbanSim counter-insurgency command simulation with the Generalized Intelligent Framework for Tutoring (GIFT). We provided an overview of the process and potential in employing GIFT to augment a training application with new capabilities for learner modeling and support. The work presented here is part of a larger project aimed at developing metacognitive tutoring functionalities for GIFT to enhance students' future learning and problem-solving abilities. Our future work includes collecting data from students using UrbanSim, performing a systematic study of the strategies they employ and their sources of confusion, and using the insight obtained from this study to identify opportunities for providing feedback and scaffolding in our GIFT tutor for UrbanSim. A study of strategies at the cognitive and metacognitive levels may require us to build an extended task model of the COIN operations that are relevant to the UrbanSim scenario. We will also work toward implementing the design recommendations that we discussed in the previous section.

Acknowledgements. This work has been supported by an ARL research grant through funding from ONR grant #W911NF-14-2-0050. We wish to thank our ARL sponsors Drs. Robert Sottilare and Benjamin Goldberg, and Lt. Col. Kenric Smith with their help in understanding and interpreting the GIFT and UrbanSim COIN environments. We would also like to thank Nate Blomberg and Mike Hoffman from Dignitas Technologies, who provided significant technical assistance as we developed the technology necessary for connecting UrbanSim and GIFT.

References

1. Nye, B.D., Sottilare, R.A., Ragusa, C., Hoffman, M.: Defining Instructional Challenges, Strategies, and Tactics for Adaptive Intelligent Tutoring Systems. In Sottilare, R., Graesser, A., Hu, X., Goldberg, B. (eds.): Design Recommendations for Intelligent Tutoring Systems (Vol. 2). U.S. Army Research Laboratory, Orlando, FL (2014)
2. Sottilare, R., Graesser, A., Hu, X., Holden, H. (eds): Design Recommendations for Intelligent Tutoring Systems (Vol. 1). U.S. Army Research Laboratory, Orlando, FL (2013)
3. McAlinden, R., Durlach, P., Lane, H., Gordon, A., Hart, J.: UrbanSim: A game-based instructional package for conducting counterinsurgency operations. In: Proceedings of the 26th Army Science Conference, Orlando, FL (2008)
4. Hu, X., Cai, Z., Craig, S.D., Wang, T., Graesser, A.C.: AutoTutor Lite. In: Proceedings of the 14th International Conference of Artificial Intelligence in Education, pp. 802 (2009).
5. Segedy, J.R., Kinnebrew, J.S., Biswas, G.: The Effect of Contextualized Conversational Feedback in a Complex Open-Ended Learning Environment. Educational Technology Research and Development, 61, 71-89 (2013)

Opportunities and Challenges in Generalizable Sensor-Based Affect Recognition for Learning

Jonathan P. Rowe[1], Bradford W. Mott[1], and James C. Lester[1]

[1]Center for Educational Informatics, North Carolina State University, Raleigh, NC 27695
{jprowe, bwmott, lester}@ncsu.edu

Abstract. Recent years have witnessed major research advances in sensor-based affect recognition. Alongside these advances, there are many open questions about how effectively current affective recognition techniques generalize to new populations and domains. We conducted a study of learner affect with a population of cadets from the U.S. Military Academy using a serious game about tactical combat casualty care. Using the study data, we sought to reproduce prior affect recognition findings by inducing models that leveraged posture-based predictor features that had previously been found to predict affect in other populations and learning environments. Our findings suggest that features and techniques, drawn from the literature but adapted to our setting, did not yield comparably effective models of affect recognition. Several of our affect recognition models performed only marginally better than chance, and one model actually performed worse than chance, despite using principled features and methods. We discuss the challenges of devising generalizable models of affect recognition using sensor data, as well as opportunities for improving the accuracy and generalizability of posture-based affect recognition.

Keywords: Affect Recognition, Posture, Microsoft Kinect, GIFT

1 Introduction

Affect is instrumental to learning. Students' affective experiences shape their learning behaviors and outcomes, and vice versa. Growing recognition of this relationship has led to the emergence of work on affect-enabled learning technologies, which endow educational software with the ability to recognize, understand, and express affect. Several affect-enabled learning technologies have been developed in recent years, spanning a broad range of domains, including computer science education [1], reading comprehension [2], mathematics [3], and computer literacy [4]. Although these bespoke affect-sensitive systems have yielded promising results, there are many open questions about whether existing affect recognition techniques generalize to new domains, populations, and settings.

Recent work on sensor-based affect recognition holds promise for yielding generalizable models. Because sensor-based models typically do not rely on features that are specific to particular learning environments, in principle, they should port across domains and settings. Sensor-based affect recognition models have been devised for a range of modalities, including facial recognition, gaze tracking, speech analysis, physiological signals (e.g., heart rate, electrodermal activity), hand gesture, and posture [5]. In this work, we focus on posture-based affect recognition, which has shown promise for its capacity to predict student affect [1, 3, 4]. Motion sensors, such as Microsoft Kinect, can be used to gather rich data streams about posture, they are relatively low-cost, and they are increasingly getting integrated into mainstream computers [6]. By modeling these rich data streams with machine learning techniques, posture-based affect recognition models have been induced that can effectively predict participants' affective self-reports, as well as expert judgments of affect gleaned from freeze-frame video analyses [1, 3, 4].

In this paper, we summarize our work on posture-based affect recognition with the Generalized Intelligent Framework for Tutoring (GIFT). In collaboration with Teachers College Columbia University and the U.S. Army Research Laboratory, we conducted a study of learner affect with cadets from the U.S. Military Academy (USMA) using a serious game for learning tactical combat casualty care skills. Using this study data, we sought to reproduce prior affect recognition findings, leveraging posture-based predictor features that had previously been found to predict affect in other populations and learning environments. However, our results indicated that the same features and techniques, adapted to our setting, did not yield comparably effective models. Our affect recognition models performed only marginally better than chance, and in fact, one model actually performed worse than chance. We discuss the challenges of devising generalizable models of affect recognition using sensor data, and describe opportunities for improving the predictive accuracy of posture-based affect recognition models.

2 Posture Sensor-Based Affect Recognition

Several research labs have investigated multimodal affect recognition in learning environments over the past decade. Our research on generalizable sensor-based affect recognition is strongly influenced by this work. To date, posture-based affect recognition models have been induced with data from pressure-sensitive chairs [3, 4], as well as motion sensors, such as Microsoft Kinect [1]. These two data streams, drawing from distinct types of sensors, are superficially different, but can be distilled into analogous predictor features that have similar relationships with affective states such as engagement, boredom, frustration, and confusion. Features can be distilled from both types of data to indicate leaning forward, leaning backward, sitting upright, and fidgeting. We summarize several representative studies that have utilized these types of features to recognize learner affect, and that have influenced our own work.

D'Mello and Graesser utilized posture data from the Body Pressure Measurement System (BPMS) to predict judgments of student affect during learning with AutoTutor [4]. The BPMS is a pressure-sensitive system that is comprised of a grid of sensing elements placed across a chair's seat and back. In their study, participants were video recorded, and several judges analyzed the video using freeze frame analysis in order to code participants' affective states retrospectively. Using this data, D'Mello and Graesser induced a series of emotion-specific binary logistic regression models, each distinguishing a particular affective state from neutral, using 16 posturebased features as predictors. Their findings indicated that the models, averaged across judges, explained approximately 11% of the variance in affective state, with findings in line with an attentive-arousal theoretical framework. Specifically, affect such as delight and flow coincided with forward leaning, boredom coincided with a tendency to lean back, and states such as confusion and frustration coincided with an upright posture.

Cooper et al. used a suite of sensors to collect data on student affect in Wayang Outpost, an ITS for high school geometry [3]. The sensors included a skin conductance bracelet, pressure sensitive mouse, pressure sensitive chair, and mental state camera, which provided data on student posture, movement, grip tension, arousal, and facial expression. The pressure sensitive chair was a simplified version of the sensing system utilized by D'Mello & Graesser [4], utilizing a series of six forcesensitive resistors distributed across the seat and back of a seat cover cushion. Data from these channels was distilled into predictor features to predict students' emotion self-reports, which were queried every five minutes throughout the learning interaction. The posture-based features included net change in seat and back pressure between the current timestep and previous timestep, and a feature indicating whether the student was leaning forward or not. Step-wise linear regression models were induced to predict students' emotion self-reports. Results indicated that posture-based features were significantly predictive of self-reported excitement during learning, although they were not part of the best-performing models for other emotional states.

Grafsgaard et al. have investigated postured-based affect prediction using Microsoft Kinect sensors with an intelligent tutoring system for introductory programming [1]. Posture features were distilled from depth image recordings by tracking the distance between the depth camera and the participant's head, upper torso, and lower torso. The features included discretized distance indicators, such as near, mid, and far head positions, each determined by whether the tracked head point was closer or farther from the median head position by one standard deviation. In addition, a postural movement feature was distilled to label occasions where the average amount of acceleration of the head tracking point was greater than the population average over a one-second window. The posture-based predictor features were combined with features distilled from other multimodal streams to induce multiple regression models for predicting students' retrospective self-reports of engagement and frustration. Findings indicated that posture features were predictive of both self-reported affective states: leaning forward was predictive of both higher engagement and higher frustration, and postural movement was associated with increased frustration and reduced learning.

Building upon this foundation, we set out to distill similar predictor features from the data collected at USMA, and apply similar machine learning methods, to produce affect recognition models for predicting field observations of affect.

3 Kinect-Driven Affect Recognition in GIFT

We collected learning and affect data from 119 USMA cadets as they used the vMedic serious game environment for learning tactical combat casualty care skills. In vMedic, the learner adopts the role of a combat medic who must properly treat and evacuate one (or several) of her injured fellow soldiers by following standard medical procedures within the game environment. All participants completed the same training module, which was managed by GIFT. The training module consisted of a pre-test, a brief PowerPoint on tactical combat casualty care, four training scenarios in vMedic, and a post-test.

Each participant was assigned to a research station that consisted of an Alienware laptop, a Microsoft Kinect for Windows sensor, an Affectiva Q Sensor, and a mouse and pair of headphones. As participants completed the study materials, a pair of field observers regularly recorded participants' physical displays of emotion. The field observers followed an observation protocol, BROMP, developed by Baker et al. [7], in which observers walked around the perimeter of the study room, discreetly recording observations of each participant's affect in a round robin sequence. The field observers coded for seven affective states: concentration, confusion, boredom, surprise, frustration, contempt, and other.

The study produced several parallel data streams, including vMedic trace data, Kinect position tracking data, electrodermal activity data, pre- and post-test response data, and field observation data. In this work, we focus on analysis of the Kinect and field observation data, which were fused into a single time-synchronized dataset. The dataset was cleaned and filtered in order to remove any Kinect-tracking glitches, as well as non-essential vertex data. Afterward, 73 predictor features were distilled, which characterized participants' postural positions and dynamics, inspired by similar features from the research literature on posture-based affect recognition. The features included summary statistics for three points tracked by the Kinect: head, top_skull, and center_shoulder. Specifically, we computed features for the current distance and depth of each vertex; the minimum, maximum, median, and variance in distance of each vertex observed thus far; the same statistics for 5, 10, and 20-second windows; several features that characterized net changes in vertex distance, analogous to the net_change features reported in [3, 4]; and sit_forward, sit_back, and sit_mid features analogous to those reported in [1, 3].

Using this feature data, we induced separate affect detectors for each emotional state using a range of machine learning techniques in RapidMiner 5.3, inclu ing J48 decision trees, naïve Bayes, support vector machines, logistic regression, and JRip [8]. The detectors were cross-validated using 10-fold participant-level cross validation. Oversampling was used to balance class frequency by cloning minority class instances in the training sets. Forward feature selection was performed to reduce the number of predictor features used in the models. We calculated Kappa and A' to assess the models' performance.

Across all of the emotions, our posture-based affect recognition models achieved an average Kappa of 0.064, and 0.521 for A' [8]. The best performing model was for boredom, which showed Kappa=0.109, A'=0.528 using logistic regression. Overall, the models performed slightly better than chance, with the exception of the surprise detector, which actually performed worse than chance, Kappa=-0.001, A'=0.493.

These results were surprisingly modest, despite our best efforts to run a carefully designed study and reproduce previously reported methods. There are several possible explanations. It is possible that BROMP labels, which are based on holistic judgments of affect over 20-second windows, are ill matched for methods that leverage low-level postural features as predictors. Previous work utilized self-reports and freeze frame video analysis, which have different tradeoffs than BROMP. Additionally, much of the work on posture-based affect recognition has taken place in laboratory settings with a single participant at a time. In our study, up to 10 participants were present, with each research station having a slightly different sensor position and orientation. This variation may have introduced additional noise to the data, which could have been problematic for the methods reported here. Further, the population of learners we used in the study, USMA cadets, showed considerable restraint in their physical expressions of affect. As such, the displays of affect via body language may have been different than those encountered in prior work, making them ill matched for the predictor features that we engineered. These findings underscore the challenges to be overcome in efforts to devise generalizable models of affect recognition.

We draw several lessons for our continued work on sensor-based affect recognition with GIFT. First, orienting Kinect sensors' position and orientation to track points on participants' lower torso could prove important for posture detection. In the present study, our sensor configuration enabled us to track only vertices on participants' upper torso and head, which may have limited the features we could distill.

Second, it would be useful to validate the Kinect vertex data recorded by GIFT against the sensor's raw depth video data. Prior work on Kinect-based posture detection directly leveraged raw depth channel data, but this method is memoryintensive and requires custom implementation of posture tracking algorithms [1]. While vertex data produced by Kinect should in principle provide the same information about posture as raw depth data, validating this fact would ensure that our findings relate to the generalizability of affect recognition techniques, and not assumptions about underlying data sources.

Third, investigating alternate machine learning techniques could prove useful for enhancing the predictive ability of posture-based predictor features. It is possible that temporal models, such as dynamic Bayesian networks, which explicitly model shifts in posture and affect, could yield improved results. Furthermore, recent work on deep learning techniques may show promise, given their capacity to perform automated representation learning. Although additional work is merited to manually engineer high-level features to match the holistic encodings of affect provided by BROMP, it would be ideal to automate this manual feature engineering process, as is one of the promises of representation learning techniques such as deep learning.

4 Conclusions

We have described work investigating the generalizability of posture sensor-based affect recognition. We collected a multimodal dataset on affect and learning with a group of USMA cadets using a serious game for tactical combat casualty care. Leveraging techniques from the affective computing research literature, we distilled a range of posture-based predictor features for modeling participants' affective states with machine learning. Our results indicated that posture-based features and models, which had previously been found to yield effective affect recognition systems, did not work as effectively on our data as had been found with other populations and learning environments. In fact, most of our affect recognition models performed only marginally better than chance, despite the use of principled features and models. Although there are several directions to investigate for enhancing our posture-based affect recognition models, the failure of existing techniques to generalize to our data is notable. These findings underscore the challenges, and opportunities, in research on affect recognition and generalizable approaches to intelligent tutoring.

Acknowledgments. The authors wish to thank the U.S. Army Research Laboratory for supporting this research. We are grateful to Vasiliki Georgoulas, Michael Matthews, and James Ness for facilitating the study at the United States Military Academy. Additionally, we wish to thank our collaborators, Ryan Baker, Jeanine DeFalco, and Luc Paquette at Teacher's College Columbia University, and Keith Brawner, Benjamin Goldberg, and Bob Sottilare from ARL.

References

1. Grafsgaard, J. F., Wiggins, J. B., Vail, A. K., Boyer, K. E., Wiebe, E. N., Lester, J. C.: The Additive Value of Multimodal Features for Predicting Engagement, Frustration, and Learning during Tutoring. In: Proceedings of the 16th ACM International Conference on Multimodal Interaction, pp. 42–49. (2014)
2. Mills, C., Bosch, N., Graesser, A., Mello, S. D.: To Quit or Not to Quit: Predicting Future Behavioral Disengagement from Reading Patterns. In: Proceedings of the 12th International Conference on Intelligent Tutoring Systems, pp. 19–28. (2014)
3. Cooper, D. G., Arroyo, I., Woolf, B. P., Muldner, K., Burleson, W., Christopherson, R.: Sensors model student self concept in the classroom. In: Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization, pp. 30–41. (2009)
4. Mello, S. D., Graesser, A.: Mining Bodily Patterns of Affective Experience During Learning. In: Proceedings of the 3rd International Conference on Educational Data Mining, pp. 31–40. (2010)
5. D'Mello, S. K., Kory, J.: A Review and Meta-Analysis of Multimodal Affect Detection Systems. ACM Computing Surveys, 47(3), 43 (2015)
6. Intel. (2015, March 20). Intel RealSense. Retrieved from http://www.intel.com/realsense.
7. Baker, R. S. J. d., D'Mello, S. K., Rodrigo, M. M. T., Graesser, A. C.: Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive– affective

states during interactions with three different computer-based learning environments. International Journal of Human-Computer Studies, 68(4), 223–241 (2010)
8. Paquette, L., Rowe, J., Baker, R., Mott, B., Lester, J., DeFalco, J., Brawner, K., Sottilare, R., Georgoulas, V.: Sensor-Free or Sensor-Full: A Comparison of Data Modalities in Multi-Channel Affect Detection. Under review for the 8th International Conference on Educational Data Mining, (under review)

The Development of a Testbed to Assess an Intelligent Tutoring System for Teams

Desmond Bonner[1], Jamiahus Walton[1], Michael C. Dorneich[1], Stephen B. Gilbert[1], Eliot Winer[2], Robert A. Sottilare[3]

[1]Industrial and Manufacting Systems Engineering, 2Mechanical Engineering
[2]Iowa State University, 3004 Black Engineering, Ames, IA, 50011, USA.
[3]U.S. Army Research Laboratory- Human Research and Engineering Directorate
Orlando, Florida
dbonner, jwalton, dorneich, gilbert, ewiner}@iastate.edu,
robert.sottilare@us.army.mil

Abstract. Work has been ongoing to develop an Intelligent Tutoring System (ITS) for teams. As part of this work, we are developing a flexible, scalable, military-based set of collaborative team tasks that can serve as a “testbed” to exercise various aspects of a team ITS architecture.Warfighting teams are a core part of any operation as individual soldiers combine their skill sets and plan, coordinate and act as one entity to accomplish assigned objectives. The team ITS test bed presented in this paper uses simple team tasks to train soldiers on basic functions including observation, target detection, target identification, communication within the team and decision making under stress. The testbed allows for manipulation of various dimensions of tutor feedback, learner workload, and team size. The testbed enables researchers to systematically evaluate the effectiveness of different types of feedback on militarily-relevant training tasks.

Keywords: Team Tutoring, Team Training, Intelligent Tutoring Systems (ITSs), Generalized Intelligent Framework for Tutoring (GIFT)

1 Introduction

Work has been ongoing to develop Intelligent Tutoring Systems (ITSs) to support tailored, guided learning experiences for teams conducting collaborative tasks [1-3]. As part of this work, we have been developing a flexible, scalable, militarily-relevant set of collaborative team tasks that can serve as "testbed" to exercise various aspects of a team ITS architecture. This paper focuses on the development of a generic testbed and an effective implementation of an ITS for training team tasks which can serve as a model for future ITSs. While work has been previously conducted in this area (see section 2), the work which is described in this paper differs as it attempts to remove humans from the tutor role completely, seeks to encourage proper performance while learners are performing several sub-tasks within a larger one, and accomplish both goals while simultaneously applying them to two or more individual learners concurrently within a collaborative team setting.

There is a need for effective team training in the military to match the tasks conducted by military teams in the operational environment. It is important that tailored training be easy to distribute while minimizing cost [4]. Tailored training through the convergence of ITSs and Virtual Reality (VR) training (e.g., serious games and virtual simulations) is emerging to become part of the Army's plan for the 21st Century soldier competencies [4,5]. VR can simulate a combat zone and allow inexperienced soldiers to learn how to react to high-stress situations without exposure to actual harm. In a virtual environment, random events can occur by the trainer's design, which mimic events such as sniper attacks, improvised explosive devices (IEDs), and hostile civilian environments. The goal for the military application of VR is not only to expose soldiers to a broad spectrum of potential environments, but also effectively train soldiers by providing tailored instruction and feedback [5]. The result is more efficient training and shorter time to reach competency.

An ITS is a computerized learning environment that incorporates content from a specific domain (e.g. military training) to provide instruction through the use of feedback and immediate interaction based on an individual learner's rate of comprehension [6]. ITSs attempt to play the role of a trainer or instructor in a training simulation. However, capturing the expertise of a human trainer is difficult. The most crucial element in training is the experience of the trainer, usually a Non-Commissioned Officer (NCO), which is shared with soldiers [7]. Beyond individual training, the military trains teams of soldiers to work together to accomplish mission goals. Military teams are capable of achieving goals that cannot be accomplished by an individual warfighter alone. Thus, the trainer is responsible for enhancing the performance and learning of multiple soldiers.

A human trainer is most effective when giving one-on-one training or tutoring [8]. The goal of ITS development was to find a tutor that was just as effective as one-to-one tutoring as it is the most effective form of education. Students who receive one-to-one tutoring perform better than those who receive conventional group education [9]. Most students have the potential to reach a high level of learning and human one-to-one tutoring allows them the opportunity to reach their potential. However, only until recently, ITS's were solely focused on individual tutoring [10]. The challenge is to make ITS training effective for teams. Developing and testing ITS for effective team training is vital to the success of military operations. Due to the increasing complexity of missions which include specific tasks, the timing and characteristics of feedback that teams receive during training is crucial to understanding a tutor's effectiveness in addition to its development [3].

Development of a Team ITS will extend an existing individual (or one-to-one) authoring architecture to small groups. Our goal is to develop an architecture for authoring team ITSs using VR and the authoring capabilities of the General Intelligent Framework for Tutoring (GIFT) [11]. This will require a test bed to assess the effectiveness of the tutor. The testbed needs to be flexible and scalable so that it can be adapted to explore different teaming variables, such as the elements and dimensions of team-based feedback [2, 12].

To develop a team training testbed, the collaborative team task of joint reconnaissance ("recon") was chosen based on its ability to test various dimensions of feedback, and its scalability with respect to workload and team size. The next section describes related work that informed the development of the testbed. The subsequent section details the generic Recon Task Testbed developed to exercise a team tutoring architecture. Finally, an initial implementation is described that tests two of the many dimensions of feedback: public vs. private, and team vs. individual feedback.

2 Related Work

Several areas of research informed the development of the Recon Task Testbed. Team training in the military and the development of individual ITSs has formed the basis of the collaborative tasks included within the Testbed. Research on the types of feedback in training scenarios was reviewed extensively. Finally, the authoring tool that is being extended from individuals to team tutors is briefly introduced. This research supports U.S. Army training objectives [5].

One of the goals for the Army is to maintain a tactical edge over potential threats through the ability to learn faster [5]. In order for teams to learn faster it is necessary for their training to be adaptive. The military is headed towards more effective training by becoming less dependent on lengthy PowerPoint slides for soldier comprehension [5]. When using an excess of PowerPoint slides to present important information students will be less engaged and unlikely to grasp material [13]. When the time comes to apply the material in field training, the learner's earlier low engagement may reflect performance. With VR training, students can be exposed to material and apply it simultaneously.

Applying VR with an ITS has been explored in previous work [4,14]. ITSs have been more effective for learning than traditional training which takes place in classrooms [6]. It reduces the time required for learning and in some cases is less costly than conventional learning. ITSs such as SimStudent predict future behavior from students by looking at previous behavioral patterns and therefore can reduce learning time [15]. It has been difficult to successfully apply what works in individual ITSs to a Team ITS [10]. Team training requires a higher expenditure of flexibility and energy in regards to authoring ITSs in addition to the human trainer. Some tutors have been created in order to assist human trainers with facilitating collaborative learning and team training such as the Advanced Embedded Training System (AETS) [16]. With AETS, the workload for the human trainer required for successful tactical team training was reduced [16].

Teams are usually made up of individuals who differ in competency, content comprehension, and skill levels. Also, team interaction is another factor which individual tutors do not have to consider. Work from Suh and Lee address the complexities of team collaborative work through an asynchronous text system called the Extensible Collaborative learning Agent (ECOLA) [17]. In their work, they go on to describe challenges such as complex educational elements which exist in collaborative systems. Specifically, feedback and the method which it is distributed can influence a team. According to Billings, feedback generally improves performance [18]. Additional characteristics of a team including how the team reacts to feedback may determine its success or failure before an assessment task even begins [1]. Team feedback has many dimensions [2]: subject (individual, team), target (public, private), timing (immediate, after), type (proactive, reactive), specificity (generic, specific), tone (positive, negative), and style (collaborative, competitive). These aspects can be effectively tested in an ITS authoring environment by using GIFT.

GIFT is a modular computer-based ITS which has three primary functions which include authoring, instructional management and evaluation of ITSs. GIFT's authoring goals are to decrease effort for creating tutors by providing aid in organizing knowledge, supporting good design principles, and leveraging open source solutions [19]. Instructional manager goals for GIFT are to integrate pedagogical best practices in ITS created from the platform. The effectiveness evaluation construct's purpose is to allow researchers to evaluate whole ITSs or their component tools and methods of ITS technologies [19]. GIFT was developed for use with individual training. The project on which this paper is based has the goal to extend GIFT to team ITSs. A team architecture has been proposed [3]. The Recon Test Bed has been developed to test that architecture.

3 Testbed Development

The Recon Testbed is based on the collaborative team task of reconnaissance, and requires several military skills. In the military, communication is key to mission success, especially for security purposes. There are four types of security operations. They include Screening, Guarding, Covering, and Area Security [20]. The Recon Scenario is derived from Area Security as it involves reconnaissance in support of various assets. Specifically it resembles aspects of patrolling. In patrolling, Observation posts are used to provide security to a platoon [7]. Within the task, users perform the five fundamentals of all security related missions. These include: orient the main body, perform continuous reconnaissance, provide early and accurate warnings, provide reaction time and maneuver space, and maintain enemy contact [7]. How well users execute these fundamentals during the task will partially determine the feedback that is received.

Feedback in teams has many dimensions (see Section 2). It is the goal of the testbed to enable experimenters to vary these dimensions as needed to test the effectiveness of team feedback. In addition, the testbed must allow the experimenter to manipulate the task load (workload) of the participant. This can be done by changing the rate at which events occur.

The recon task itself, built in VBS2, is meant to serve as the testbed for these dimensions. In conducting the task, users are exposed to various military scenarios such as observation, fields of fire, and communication within a fire team element. The team members (two minimum) are assigned sectors to watch. For instance, if there are four teammates on the top of a building, each may be assigned one quarter of the 360-degree field of view. Each is tasked with scanning (observing) their sector by constantly panning to see the extent of activity (target detection) in their sector. Each trainee must identity (target identification) any opposing force member that is spotted. If the threat is moving into a teammate's sector, the learner then must transfer responsibility by communicating the position to that teammate. The teammate must then acknowledge the change of responsibility back to the first teammate, thus accepting responsibility.

In the example of four team members, the initial condition of scanning is based on the 90-degree sector given to each team member. The team member must scan this sector continuously for the purpose of mimicking the actual field task and to effectively participate in the other conditions of the recon scenario. The team is given feedback according to how effectively they cover their entire area. This is relative to fields of fire and reconnaissance strategies outlined in the Army Field Manual for Infantry Platoons and Squads [7].

Figure 1 illustrates two teammates (BLUFOR) each monitoring a 90-degree sector. Participants are responsible for tracking all targets (OPFOR) and ignoring any distractors (civilians). When a target approaches the sector border in the center, the participant must alert the team member who has responsibility for that sector. Workload can manipulated by changing the number of enemies/civilians, the speed by which they move, the similarity of their appearance, and the rate by which they appear/disappear.

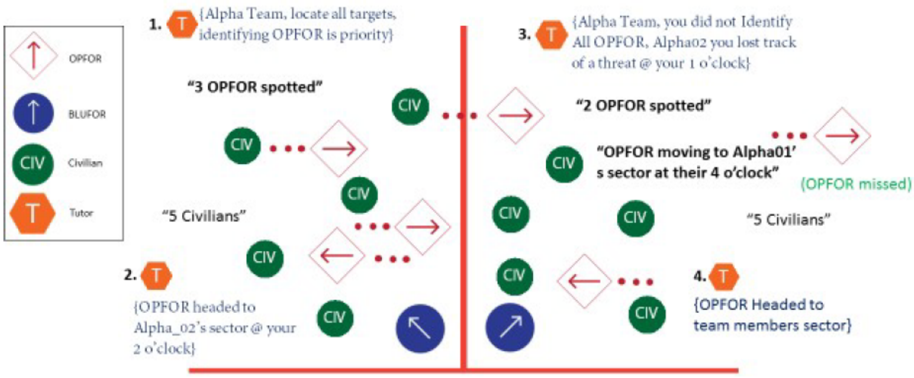

Fig. 1. Example of a recon task in which two team members scan a 180-degree field.

The dimensions of feedback can be varied in the task by changing the content or delivery of the ITS feedback. Table 1 describes how feedback dimensions can be manipulated in the Recon Testbed to test the effectiveness of team feedback.

Table 1. Dimensions of Feedback

Dimension	Levels	How realized in Recon Testbed
Subject	*Individual, Team*	Tutor provides feedback about an **individual** team member or **entire** team
Target	*Public, Private*	Tutor provides feedback to either a single person (**private**) or team (**public**)
Timing	*immediate, after, omitted*	Feedback occurs based on patterns or task effectiveness **during** the task, or **after** overall the grade or rating is given. Feedback is **omitted** when an error is committed, but is not sufficiently important to interrupt training to provide immediate feedback or to be included in the After Action Review.
Type	*Proactive, reactive*	**Proactive**: feedback before a learner makes error, **Reactive**: Feedback after a learner makes an error
Specificity	*Generic, specific*	**Generic:** "*Good Job Soldier*" **Specific**: "*You missed an OPFOR located at 7 o'clock*"
Tone	*Positive, negative*	**Positive:** "*...you might want to try…*" **Negative**: "*...your poor performance is hurting the team*"
Style	*Collaborative, Competitive*	**Collaborative**: "Slow down scanning to help team…" **Competitive**: "Your performance is worse than Joe."

4 Initial Implementation and Future Work

The first implementation will study two dimensions of feedback: Access (public vs. private) to feedback, and target (group vs. individual) feedback. For example, the feedback is given to a single person in the private condition while the entire team is given feedback in the public setting. Individual and Group feedback refers to whom the feedback is about (one person's actions or the team's efforts). Table 1 describes the tasks of each learner when monitoring their sector. The team tutor will be the basis of experiments to test the effectiveness of different types of team ITS feedback.

Table 2. Tasks performed in the initial Recon Testbed by each learner.

Task	Description
Scanning	The Learner rotates their viewpoint within the 180 degree sector. Learner must cover the entire 180 continuously throughout the task
Identify	The learner presses a key whenever they spot a new OPFOR avatar. This must be done quickly with 10 seconds of the OPFOR becoming visible
Transfer (informing)	When an OPFOR avatar is close to moving into a teammate's assigned sector, the learner must inform the team member.
Transfer (confirming)	Learner must confirm transfer of responsibility for the OPFOR moving into their sector from the teammate who initiated the transfer process.

Beyond the initial study, we plan to expand the Recon Testbed significantly. Currently, the testbed allows for the manipulation of feedback dimensions that enables researchers to systematically test the effectiveness of different types of feedback on training. The testbed is scalable and flexible, allowing for different sizes of teams, and varying levels of task load, which can be altered in the future. By including these features, the testbed will provide a platform to study several aspects of military-relevant team training.

References

1. D. Bonner, S. Gilbert, M.C. Dorneich, S. Burke, J. Walton, C. Ray, & E. Winer, (2015, February). Taxonomy of Teams, Team Tasks, and Tutors. In *Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium*.
2. J. Walton, M.C. Dorneich, S. Gilbert, D. Bonner, E. Winer, & C. Ray (2015, February). Modality and Timing of Team Feedback: Implications for GIFT. In *Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium*.
3. E. Winer, J. Holub, T. Richardson, M. Hoffman, S. Gilbert, & M. Dorneich (2015). "Characteristics of a Multi-User Tutoring Architecture software architecture," In. R. Sottilare (Ed) *2nd Annual GIFT User Symposium*. Pittsburgh, PA, June 12-13. *Army Research Laboratory*, Orlando, Florida, 2015. ISBN: 978-0-9893923-4-1
4. J. Stevens, P. Kincaid, & R. Sottilare (2015). Visual modality research in virtual and mixed reality simulation. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*.
5. M. E. Dempsey, (2011). The US Army learning concept for 2015 (TRADOC Pam 525-8-2). Washington, DC: Department of the Army HQ, US Training and Doctrine Command.
6. A. C. Graesser, M. W. Conley, & A. Olney, (2012). Intelligent tutoring systems.

Developing an Experiment with GIFT: Then and Now

Anne M. Sinatra, Ph.D.[1]

[1]U.S. Army Research Laboratory – Human Research and Engineering Directorate
anne.m.sinatra.civ@mail.mil

Abstract. The Generalized Intelligent Framework for Tutoring (GIFT) is a domain-independent open-source intelligent tutoring framework. In the past new versions of GIFT were released every 6 months, and currently, officially tested versions of GIFT are released every 9 months. Each new version of GIFT includes additional capabilities and functionalities. In the current paper and presentation, the "Logic Puzzle Tutorial" course that was developed in GIFT 2.5, and has been included with releases of GIFT since version 4.0 will be discussed. The presentation will describe the rationale and methods behind the course's development, and discuss different approaches that might have been used with the features that are present in GIFT today.

Keywords: Adaptive Tutoring, Intelligent Tutoring, Experimental Design, Course Development, Generalized Intelligent Framework for Tutoring, Psychology

1 Introduction

The Generalized Intelligent Framework for Tutoring (GIFT) is an open-source domain-independent intelligent tutoring framework [1]. Since GIFT is domain independent it offers great flexibility in the types of tutors and experiments that can be developed with it. While the development of adaptive tutoring systems is a primary objective of GIFT, it was also designed to be used as a testbed and for analysis purposes. Experiments can and have been developed and run using GIFT [2,3]. GIFT provides opportunities to create experiments that use adaptive feedback/assessment, and experiments that do not. In fact, GIFT is very useful as a mechanism to run traditional experiments in the area of psychology [4]. One such experiment was run as part of a Post Doctoral fellowship with Army Research Laboratory to investigate the impact of self-reference and context personalization on computer-based tutoring [3,5]. The skill that was taught to individuals was deductive reasoning, which was done through an interactive logic puzzle tutorial. The current paper discusses the development of the logic puzzle tutorial, and the different approaches that may have been taken had the current features of GIFT been available at the time.

1.1 Logic Puzzle Tutorial Experiment

The "Logic Puzzle Tutorial" course has been included with GIFT software releases since GIFT 4.0 in November 2013. This tutorial was originally developed for use in an experiment to examine the impact of self-reference on learning deductive reasoning skills and completing logic puzzles. The description and results of the original experiment are available in the form of an Army Research Laboratory technical report [5]. In the full experiment there were 3 versions of the logic puzzle tutorial. All of the versions were identical except for the names that were included in the puzzles and learning material. The names that were included were determined based on the condition, and the names that the participants were asked to type into the program. In the self-reference condition, the participant entered his or her name, and the names of 2 friends. In the popular culture condition, the participants were prompted to enter specific names of characters from the Harry Potter series. In the baseline condition, participants were asked to enter 3 provided names that were not common for their age group (based on birth name data). See Figure 1 for a screenshot comparison of the popular culture and baseline conditions.

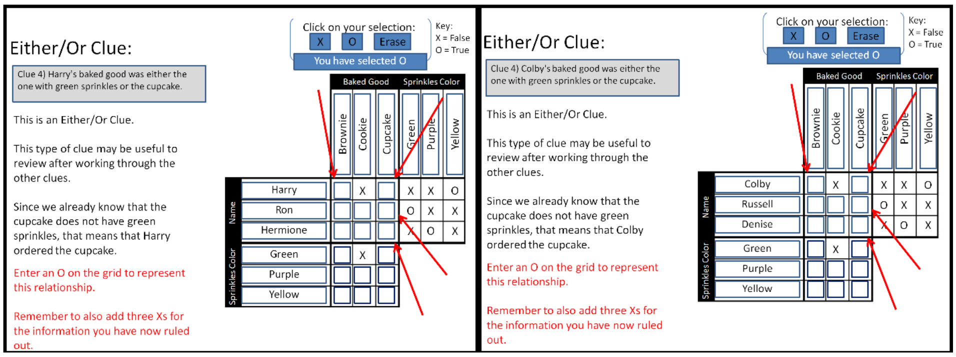

Fig. 1. Screenshots that demonstrate the manipulation of interest in the original logic puzzle tutorial experiment. Note that the names present in the puzzles and clues are different, with the Popular Culture condition on the left, and the Baseline condition on the right.

In the experiment, adaptive tutoring and feedback were provided to participants through a logic puzzle tutorial created in PowerPoint with Visual Basic for Applications (VBA). GIFT provided the interface that participants used for the study, presented surveys, opened and closed the PowerPoint based tutorial, launched web-page based questionnaires, and connected to a Q-sensor for physiological data collection. The original course was developed in GIFT 2.5, which was an experiment-based version of the November 2012 release of GIFT 2.0. Despite being developed in GIFT 2.5, the course is still compatible with current versions of GIFT (at the time of writing the most recent version is GIFT 2014-3X, which was released in December 2014).

1.2 Version of the "Logic Puzzle Tutorial" course included with GIFT

The released version of the "Logic Puzzle Tutorial" is a slightly modified version of the baseline condition tutorial from the original experiment and includes the names that were used in the experimental version. In this version, the tutorial automatically has the names present in it as opposed to prompting the user to enter them as in the experiment. The released version of the tutorial course includes a subset of the questionnaires and question based knowledge assessments that the participants answered. In the full experiment, after the completion of the tutorials the participants answered multiple-choice assessments, engaged in solving an "easy" puzzle and then a "difficult" puzzle. The released version of the course only includes the "easy" puzzle. See Figure 2 for a screenshot of the "easy" puzzle that is included with GIFT releases. Unlike the tutorial portion of the course, the "easy" puzzle does not include any adaptive feedback to the participants. However, the answers that are provided by the participant are saved to an external excel file for future analysis. There are two output files of interest for the researcher: 1) output of the surveys/questionnaires that can be accessed through GIFT's Event Reporting Tool (ERT), and 2) Excel output of the puzzle which is saved in the Domain folder associated with the "Logic Puzzle Tutorial" course.

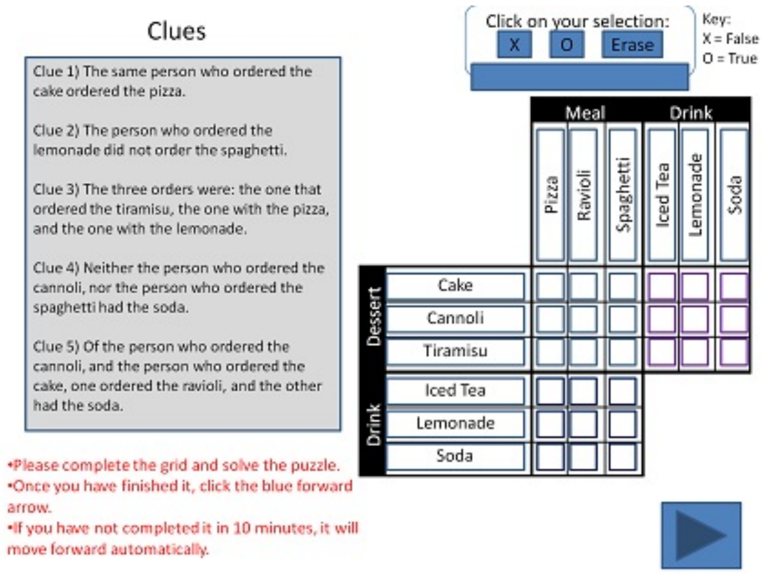

Fig. 2. Screenshot of the "easy" logic puzzle that is included with the "Logic Puzzle Tutorial" course in GIFT.

2 Tools used in Course Development: Then and Now

GIFT contains a suite of Authoring Tools that can be used for course development. The tools that were used in the development of the original "Logic Puzzle Tutorial" course/experiment were the Course Authoring Tool (CAT), Sensor Configuration Authoring Tool (SCAT), and Survey Authoring System (SAS). Since the adaptive feedback occurred within the PowerPoint tutorial, a placeholder Domain Knowledge File (DKF) was used that did not result in adaptive feedback provided directly by GIFT. As GIFT has continued to develop, many of GIFT's tools have been updated and have new functionalities in their current versions.

2.1 Course Authoring Tool (CAT) and GIFT Authoring Tool: Then and Now

The primary tool used for the development of the "Logic Puzzle Tutorial" was the CAT. The CAT allows the author to create a course flow that includes the order of guidance, training applications (e.g., PowerPoint), and surveys that the participant receives. Once design decisions have been made about the course and the components have been created, the CAT is where the transitions and flow of the course are specified. Figure 3 is a screenshot of the original "Logic Puzzle Tutorial" course loaded in the CAT. Note the linear structure of the elements, and the nodes that can expand to provide more detail.

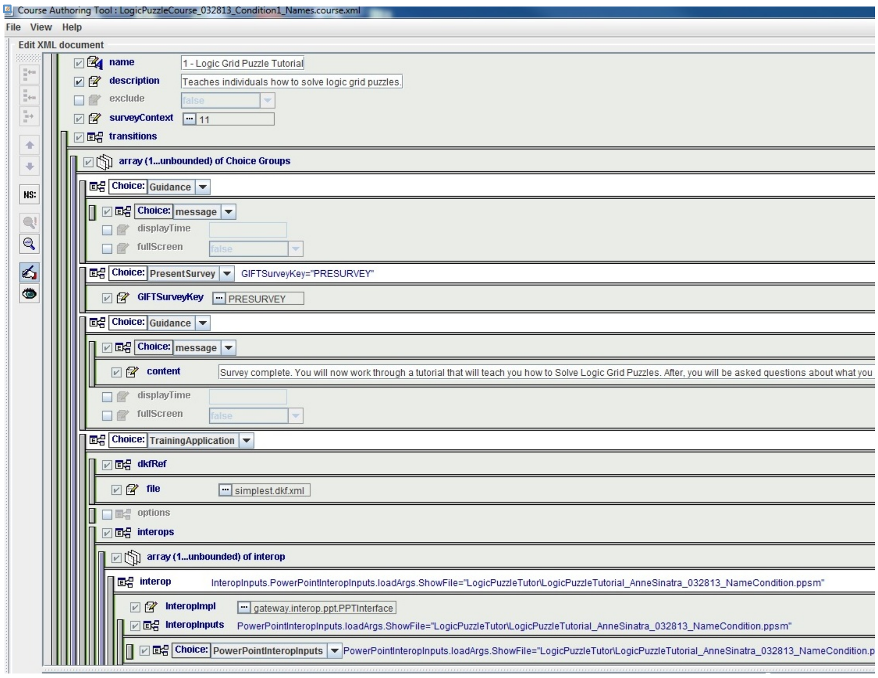

Fig. 3. Screenshot of the Course Authoring Tool that was used to create the "Logic Puzzle Tutorial" course.

The original XML (Extensible Markup Language) editor based CAT is still included with current releases of GIFT. However, an additional GIFT Authoring Tool (GAT) has been designed to allow an author to perform the same functionality in a more user-friendly interface. The same functionalities and course elements can be created using the GAT, but the interface is more straightforward and uses drop down menus that are closer matches for a general user's mental model than an XML editor based tool. A screenshot of the GAT with the "Logic Puzzle Tutorial" course loaded in it can be seen in Figure 4. While the redesign of this tool would not have impacted the design of the original course, it is expected that it would have led to a faster understanding of how to create the GIFT course.

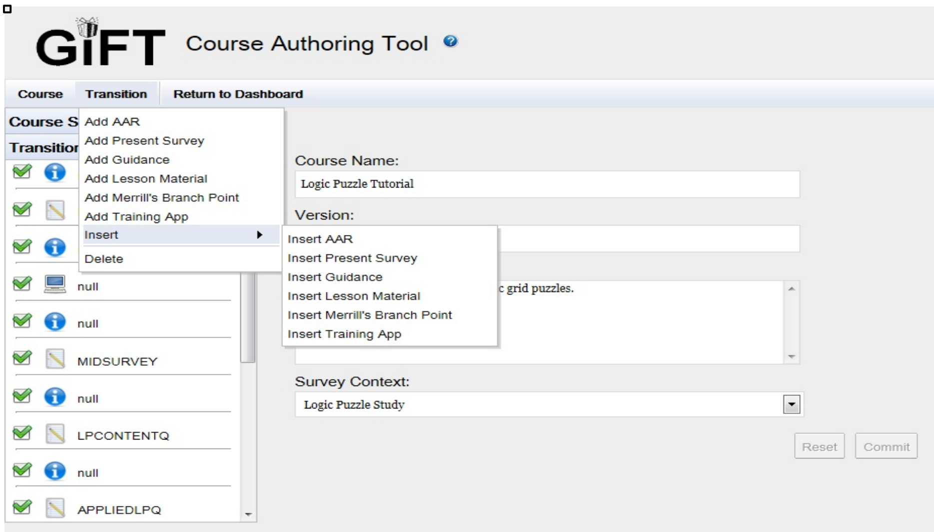

Fig. 4. The "Logic Puzzle Tutorial" course loaded in the GIFT Authoring Tool.

2.2 Survey Authoring System (SAS): Then and Now

The SAS was heavily used in the design of the "Logic Puzzle Tutorial" course. Many surveys including multiple-choice and multiple answer questions were created for use in the course. All of these surveys are available in releases of GIFT from version 3.0 to present, and many of these surveys are referenced within the "Logic Puzzle Tutorial" course. See Figure 5 for a screenshot of the SAS.

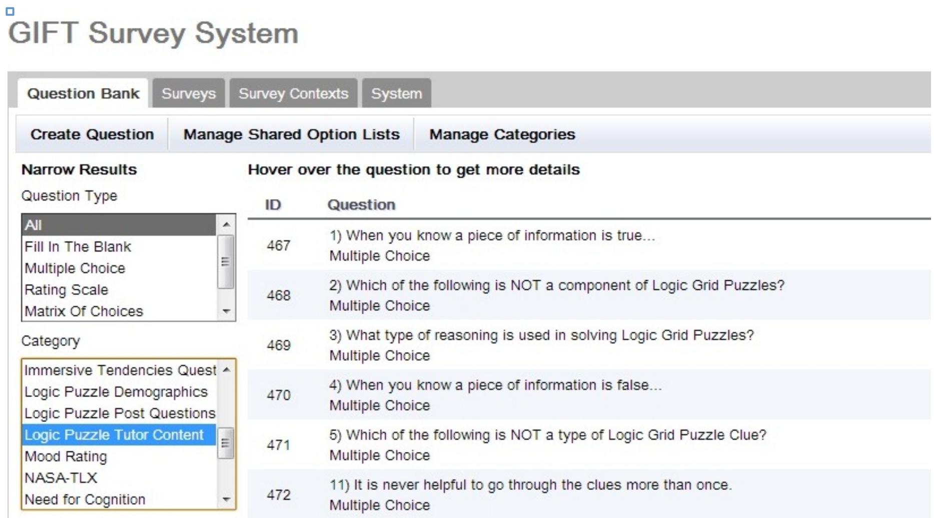

Fig. 5. Screenshot of GIFT's Survey Authoring System and a selection of questions associated with the "Logic Puzzle Tutorial" course.

The primary functions of the SAS have remained stable since the design of the original "Logic Puzzle Tutorial" course. However, there are now additional features that would be used. In the design of the original course the outputs of the questions were not automatically scored. Part of the reasoning behind this decision was that many of the scoring features were still in development at the time. Now the scoring features are stable and well documented in GIFT's doc files. Additionally, course examples that use scoring can now be viewed and examined by authors to understand the scoring functionality. Weights can be assigned to the answers in the creation of questions, and surveys can be scored. Additionally, with the development of the Engine for Management of Adaptive Pedagogy (EMAP), question banks can now be created that are associated with specific concepts that the learner can be assessed on. The grading of surveys can now influence remediation that the individual learner is given. The functionality provided by the EMAP may have influenced the design of the logic puzzle tutorial experiment if it was created today, and may have ultimately led to a different experimental design. See Figure 6 for a screenshot of a survey context with a question bank in the SAS that is associated with the functionality of the EMAP. The development of the EMAP has been documented in the literature, which can be referenced for further reading [6,7].

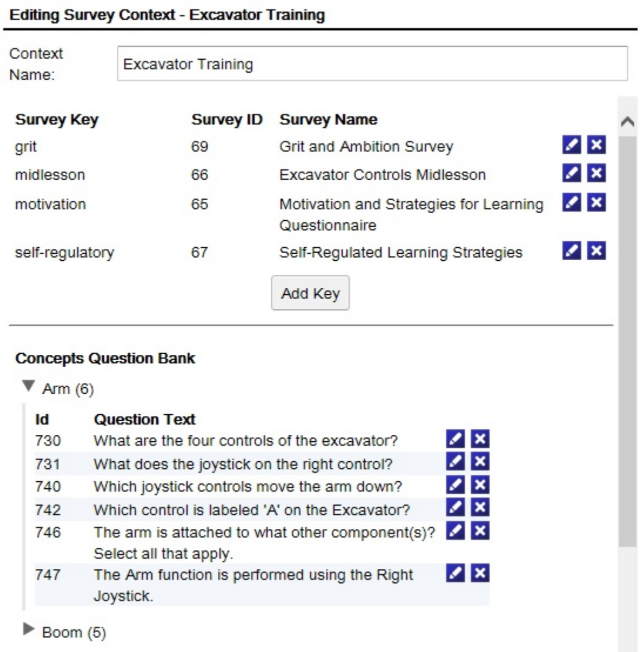

Fig. 6. Screenshot of a question bank associated with a Survey Context in GIFT 2014-2's SAS

2.3 The Sensor Authoring Tool (SCAT): Then and Now

The SCAT has remained fairly constant since the development of the "Logic Puzzle Tutorial" course. Like the CAT, it is an XML editor based tool. Default configurations for specific sensors are included with GIFT and authors can change the reference for the sensor configuration that will be used when they run GIFT. The sensor configuration is not linked directly to a course, but is used in all instances of the installation of GIFT unless it is adjusted between learners. Future versions of GIFT are expected to move toward making connections between the sensor configuration and the specific course that has been designed and run.

3 The Future

GIFT has gone through many iterations through the years, and at each point has added additional functionality and features. More additions and adjustments are expected as GIFT moves forward and in new directions, such as the cloud. One of the current goals of GIFT is to improve usability, which will make current and future features more understandable to all GIFT users. The "Logic Puzzle Tutorial" course which exists in GIFT is an example of using GIFT for an experiment. While it does not include adaptive elements based in GIFT, it offers a demonstration of how GIFT can be used for a traditional psychology experiment. The features of the more recent versions of GIFT provide more flexibility and options to individuals who will be designing experiments in the future.

References

1. Sottilare, R. A., Holden, H. K.: Motivations for a Generalized Intelligent Framework for Tutoring (GIFT) for Authoring, Instruction, and Analysis. *AIED 2013 Workshops Proceedings Volume 7* pp. 1 - 8 (2013).
2. Goldberg, B., & Cannon-Bowers, J. (2013, July). Experimentation with the Generalized Intelligent Framework for Tutoring (GIFT): A Testbed Use Case. In *AIED 2013 Workshops Proceedings Volume 7* pp. 27 – 36 (2013).
3. Sinatra, A. M. Using GIFT to Support an Empirical Study on the Impact of the Self-Reference Effect on Learning. In *AIED 2013 Workshops Proceedings Volume 7* pp. 80 – 87 (2013).
4. Sinatra, A. M. The Research Psychologist's Guide to GIFT. In *Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym2)* pp. 86- 93 (2015).
5. Sinatra, A. M., Sims, V. K., Sottilare, R. A. *The Impact of Need for Cognition and Self-Reference on Tutoring a Deductive Reasoning Skill* (No. ARL-TR-6961). Army Research 6. Lab Aberdeen Proving Ground, MD (2014).
6. Wang-Costello, J., Tarr, R.W., Cintron, L.M., Jiang, H., & Goldberg, B. Creating an Advanced Pedagogical Model to Improve Intelligent Tutoring Technologies. In *The Interservice/Industry Training, Simulation & Education Conference Proceedings* (2013).
7. Goldberg, B. What Makes an Effective Pedagogical Model? In ITS 2014 Workshop Proceedings: Pedagogy That Makes A Difference: Exploring Domain-Independent Principles across Instructional Management Research within the ITS Community pp. 1 - 9 (2014).

Adaptive Course Flow and Sequencing through the Engine for Management of Adaptive Pedagogy (EMAP)

Benjamin Goldberg[1] and Michael Hoffman[2]

[1]U.S. Army Research Laboratory, Orlando, FL
[2]Dignitas Technologies, Inc., Orlando, FL
benjamin.s.goldberg.civ@mail.mil;
mhoffman@dignitastechnologies.com

Abstract. The Engine for Management of Adaptive Pedagogy (EMAP) is the Generalized Intelligent Framework for Tutoring's (GIFT) first implementation of a domain-independent pedagogical manager. It establishes a framework within GIFT that adheres to sound instructional system design, while also providing tools and methods to create highly personalized and adaptive learning experiences. In this paper, we present the components of the EMAP, we highlight their utility when authoring an EMAP managed lesson, and we review the limitations associated with its first instantiation.

Keywords: Adaptive Instruction; Pedagogical Model; Instructional Management; Engine for Management of Adaptive Pedagogy; Generalized Intelligent Framework for Tutoring

1 Introduction

The Generalized Intelligent Framework for Tutoring (GIFT) is being developed as a domain-agnostic solution to authoring, delivering, and evaluating adaptive training solutions across an array of domains and training applications. While GIFT's initial development focused on establishing a standardized architecture for building Intelligent Tutoring System (ITS) functions to support distributed learning events, recent work has centered on extending the adaptive capabilities the framework affords. As a result, the Engine for Management of Adaptive Pedagogy (EMAP) was developed. The EMAP is based on an extensive literature review of instructional strategy focused research within computer-based training [3], and organizes its findings in a domain-independent fashion. At the moment, there are papers that highlight the literature and theory that fed the EMAPs design [3, 4] and that highlight the authoring tools and processes required for implementing its functions [5], but there is nothing that reviews EMAP interactions from the learner's perspective as it relates to event sequencing. In this paper, we present a usecase of a GIFT lesson managed by the EMAP and we review the various architectural components that make it run. We will first highlight the work that went into formalizing the EMAP, the dependencies the EMAP has with other portions of the GIFT architecture, and we present a usecase of lesson interaction and transitions managed by EMAP logic and configurations.

2 Formalizing the EMAP

The EMAP design was the resulting outcome of a collaborative project between the U.S. Army Research Laboratory (ARL) and the Institute for Simulation and Training (IST) at the University of Central Florida. Following an extensive literature review, the team selected David Merrill's Component Display Theory (CDT) as the theoretical framework to structure EMAP requirements around [3,5].

The CDT was conceptually integrated within GIFT as a domain-agnostic framework used for course construction and building guidance/remediation configurations [3]. This requires linking learner relevant information with generalized descriptors of learning content and instructional techniques, strategies and tactics. These relationships were used to establish an initial decision tree that informed real-time adaptations.

It is important to highlight the current attributes represented in a GIFT learner model and their relationship with metadata used to describe learning content. As these variables moderate EMAP configurations that are set and adapted at run-time, it is important to review how each level of data operates and what decisions they inform. For learner model data forms, these include determinations for knowledge states, skill states, affective states, and individualized traits that have been empirically found to impact learning and retention.

2.1 Learner Model Dependencies

The EMAP uses pedagogical configurations that are moderated by attributes being tracked in GIFT's learner model. These configurations are coupled to the customized value ranges of available variables supported within the architecture's standardized schema. The configurations implemented in the EMAP are based on both historical and real-time inferences across the various trait and state attribute spaces. As such, the EMAP uses information on prior knowledge along with a set of trait characteristics to personalize lesson materials across the CDT's four quadrants (i.e., Rules, Examples, Recall, and Practice) upfront, and then uses real-time assessment information on knowledge, skill, and affective states to moderate guidance, remediation, and problem selection. The goal is to establish generalized configurations that can translate across different domain spaces and varying training platforms and applications.

For knowledge and skill states, performance is monitored at an objective level. In the latest release, GIFT tracks individual learners across a hierarchy of concepts as they relate to a set of tasks within a specified domain. These concepts are established in the Domain Knowledge File (DKF), where bottom level sub-concepts (i.e., leaf nodes) are assessed against data made available by the training application itself. For each concept and set of sub-concepts, there are currently four possible state determinations: (1) above-expectation, (2) at-expectation, (3) below-expectation, and (4) unknown. Each of these representations can be associated with either a knowledge state or skill state, where this division is used to differentiate 'knowledge' from 'ability to execute'. This falls in line with the mention of Knowledge/Skills/Abilities (KSAs) defined in most doctrine and helps to make competency badging within a domain more granular. Inference procedures are performed across all concepts to determine a competency level for the domain of instruction, with values being entered as Novice, Journeyman, or Expert.

Variables based on traits found to impact learning are of importance to the EMAP. The individual traits of a learner are believed to be more stable over time and are used to set initial configurations of a lesson based on these associations. Current EMAP logic informed by traits includes motivation, self-regulatory ability, and grit. These items are not inherently tracked in the DKF, but they are used offline to configure lesson materials and sequencing when a lesson is initialized.

In terms of affect represented within GIFT's learner model, these state spaces associate primarily with data made available through sensor technologies that monitor both physiological and behavioral data sources. Affective states of interest include engagement, frustration, boredom, confusion, etc. Regardless of the state space, GIFT is very flexible with respect to affective modeling, as the researcher and/or training developer has the ability to configure what variables to track and what classifiers to apply. These classifiers are used to produce a state determination that is represented in GIFT's DKF across short-term, long-term, and predicted values. For adaptation purposes, much of the affect related information is used to adapt instruction during run-time, as this form of assessment provides insight into a learner's reactive tendencies to an event or interaction.

2.2 Metadata Dependencies

Learner model attributes are linked with generic content descriptors that the EMAP is designed to act on. This metadata is used to take domain-independent representations of pedagogical practice and associate it with domain-specific content. The metadata currently in use is based on the Learning Object Metadata (LOM [6]) standard put in place by the Institute for Electrical and Electronics Engineers (IEEE). This provides a set of high level categories (e.g., interactivity type, difficulty, skill level, coverage, etc.) and value ranges (i.e., skill level is broken down into novice, journeyman, and expert) that inform characteristics for a type of interaction. GIFT uses two authoring processes to build the EMAP linkages. First, a lesson developer needs to build metadata files for all associated content and practice materials. Next, the lesson developer must establish what learner model attributes moderate metadata selection, and what value ranges serve as strategy selection thresholds.

2.3 EMAP Course Flow Example

The following use case represents the interaction of GIFT transitions across lesson elements and materials. Each event is described in relation to the EMAP and the type of data that informs its application. The usecase is broken down by learner login and course selection; pre-lesson learner model updates and assessments; adaptive lesson delivery via a Merrill's branching; and After Action Review (AAR) and lesson completion.

Learner Login and Course Selection. When a learner interacts with GIFT to initialize a course or lesson, they are first required to login using associated IDs and passwords. Once logged in, the first function GIFT performs is checking for long-term learner model information, such as records of prior training events and any persistent trait variables being stored over time (this latter function is currently being developed). Presently, all prior training events are stored under experience Application Programming Interface (xAPI) specifications within a designated Learner Record Store (LRS) [1]. Out of the box GIFT isn't configured to use an LRS, just the SQL database we have been using for years. However the GIFT in the cloud instance will be configured to use the ADL LRS (but even that clears data out every day or so). No matter if the data is stored in either place, GIFT makes use of that information. Information related to courses taken along with performance outcomes on a concept by concept level are communicated. This information is used to recommend courses based on if any prior training events resulted in below-expectation outcomes. This is the current role xAPI plays in this process. We expect this capability to become more robust over time. Following this update, a learner is then able to select a course from GIFT's Tutor User Interface (TUI). Following this update, a learner has the ability to select their course and progress into the first transitions of a lesson.

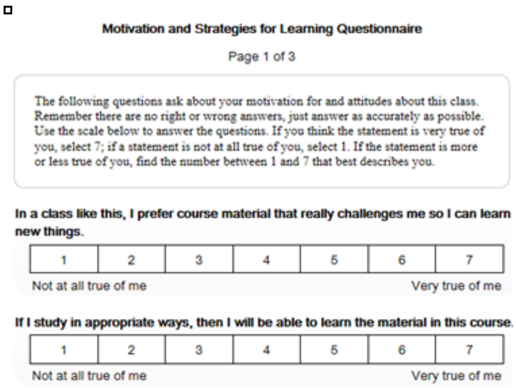

Fig. 1. GIFT Survey Interface

Pre-Lesson Updates and Assessments. Upon course initialization, GIFT references the EMAPs pedagogical configuration file to determine the trait-based variables that moderates adaptations to the lesson structure. In the current baseline, these variables include motivation, prior knowledge, self-regulatory ability, and grit. Other variables such as skill and goal-orientation can also be applied, which is the current case when a learner enters a practice quadrant of the CDT. A lesson developer has the ability to select which variables to moderate their lesson adaptations around, which impacts the first transitions experienced by a user in a new lesson. GIFT will first check an individual's persistent long-term learner model to identify any existing data. If no record is located, GIFT will administer an available survey to collect that information. This interaction is authored in GIFT's Survey Authoring System and is presented directly to the learner on the TUI (see Figure 1). Scoring rules are associated with all administered instruments, which are then used to update learner model attribute values in real-time.

GIFT then establishes learner knowledge and skill states based on associated xAPI data that exists for that domain. If no data is available, then knowledge and skill attributes are set to 'Novice'. Next, if a lesson pre-knowledge assessment is made available by the lesson developer, then the test is presented to the learner through GIFT's TUI. Based on established scoring conditions for that assessment, the learner model is updated accordingly to reflect new predicted competency levels. This information is used to bypass lesson materials on concepts that the learner has exhibited expert understanding of. Bypassing concepts is dependent on the separation of concepts not only in how they are sequenced in the course.xml but also in the content presented. i.e. if there is only 1 piece of content that covers A+B, how can either one be skipped and not the other?

Adaptive Lesson Delivery via Merrill's Branching. Once all trait-based information has been established in the learner model and all pre-test assessments have been administered, a learner is then progressed into the adaptive lesson deliver through a set of pre-defined Merrill's Branching points. This entails customized sequencing through the CDT quadrants. This interaction will be outlined through the following collection of bullet points.

- *Rules and Examples Quadrants*: Configure material around defined concepts being instructed and known attributes of the learner that match entries within the EMAP's decision tree
  - Attributes
    - Knowledge; Motivation; Self-Regulatory Ability; Grit
  - Proposed Assessments
    - Affective State: monitor learner to assess emotional and cognitive reactions
    - Behavior: monitor behavior within learning environment to assess gaming behaviors
  - No knowledge/skill updates in learner model will occur within these quadrants
- *Recall Quadrant* (Knowledge Assessment):
  - If a bank of questions for this concept has been authored within the SAS, then deliver randomized recall assessment based on EMAP configuration (configuration is defined within GIFT's Course Authoring Tool; see Goldberg et al., 2015)
    - If established scoring conditions exist, then update learner model based on assessment outcomes
      - Assumption: Only <u>cognitive knowledge</u> is updated based on performance outcomes within a survey delivered assessment within the recall quadrant
  - Guidance Configuration (currently being developed)

    - Use known attributes of the learner to configure timing and specificity dimensions
      - Question by Question Feedback vs. Following All Items
        - Attributes that may dictate this decision: Knowledge and Self-Regulatory Ability
      - General to Specific vs. Specific to General Feedback
        - Attributes that may dictate this decision: Knowledge and Grit
  - Remediation
    - If learner is reported at 'below expectation'/'at expectation' on any items (i.e. concepts), then initiate remediation loop within the defined Merrill Branch
      - Remediation path is dependent on reported cognitive knowledge state based on defined scoring logic in the Course Authoring Tool
        - For each concept:
          - If learner is scored at 'below expectation' based on scoring configuration, select that concept for Rule quadrant remediation
          - If learner is scored at 'at expectation' based on scoring configuration, select that concept for Example quadrant remediation (can be in addition to Rule quadrant remediation)
        - If there is any concept remediation needed, present the Rule remediation for all identified concepts followed by Example remediation.
          - This is where the metadata selection algorithm is used to select different content to deliver to the learner (if available).
    - Remediation ends back in Recall Quadrant
      - If items report at 'below expectation' again and there is no new content to present; then allow the learner to select the quadrant they prefer to remediate in (currently being developed).
  - If all items in the Recall Assessment are reported at 'above-expectation' then move onto Practice.
- If no questions exist for the concepts within the SAS or the author removed the recall quadrant from the branch, then move onto Practice (not currently supported).
- *Practice Quadrant* (Skill Assessment):
  - If no practice has been authored/configured, and the Recall Quadrant has been satisfied, then move onto next transition in the course file
  - If a training environment/scenario has been configured, then deliver practice materials through pre-established Gateway and DKF
  - Configure material around known attributes of the learner that match entries within the EMAP's decision tree (to be developed)
    - Attributes
      - Skill; Motivation; Self-Regulatory Ability; Grit; Goal-Orientation
    - Proposed Assessments
      - Affective State: monitor learner to assess emotional and cognitive reaction
      - Behavior: monitor learning environment to assess gaming behaviors
      - Skill: monitor performance in real-time across all identified sub-concepts based on pre-defined assessments authored around Evidence Centered Design (Stealth Assessment; [2])

    - Using established scoring conditions, update learner model based on assessment outcomes
    - Assumption: Only cognitive skill is updated based on performance outcomes within a practice environment
    - A survey authored in the SAS can also be defined as a practice environment (currently being developed).
- Guidance Configuration (currently being developed)
  - Use known attributes of the learner to configure timing and specificity dimensions
    - Number of violations before triggering guidance/feedback
      - Attributes that may dictate this decision: Skill and Self-Regulatory Ability
    - General to Specific vs. Specific to General Feedback
      - Attributes that may dictate this decision: Skill and Grit
    - Static (text or audio alone) vs. interactive (AutoTutor reflection)
- Remediation
  - If learner is reported at 'below expectation'/'at expectation' on any items, then initiate remediation loop within the defined Merrill Branch
    - Remediation path is dependent on a combination of skill and knowledge
      - If learner is novice in skill and expert in knowledge, then re-initialize practice
      - If learner is novice in skill and journeyman in knowledge, then navigate to examples quadrant
  - Remediation ends back in Recall Quadrant (currently being developed)
    - If items report at 'below expectation' again and there is no new content, then allow the learner to select the quadrant they prefer to remediate in
- If all items in the Practice Assessment are reported at 'above-expectation' then move onto next transition in the course file

This sequence of interaction will occur for all identified Merrill's Branching points authored. For instance, in a lesson that instructs across four concepts, an author can decide to break up the material across two branching points. Regardless of the number of Merrill's Branching points, once all exit criteria has been reached, then the lesson transitions into post-test assessments, after-action review and lesson completion.

Post-Lesson Assessment, After Action Review, and Lesson Completion. Upon completion of all adaptive lesson transitions across the designated Merrill's Branch points, a course developer will have the ability to administer a post-knowledge and/or post-skill assessment as a means for determining overall competency levels following lesson interventions. These interactions are intended to be void of guidance functions to determine how learners perform on their own. The outcomes are used to establish final score and attribute values for a lesson, with future development offering extended remediation events.

Assessment exercises are followed by a GIFT managed AAR used for reflective and summarization practices. It is during this interaction that a student is directed to reflect on the experience of the instructional event and their resulting performance outcomes. GIFT's current AAR capability is a web-page that reviews the objectives and concepts of a lesson taken, along with recorded performance measures for all items. A goal is to provide an interactive AAR function that utilizes technology to engage a learner in reflective exercises. Following execution of the AAR transition, the EMAP managed GIFT course is complete. At this instance, GIFT communicated xAPI data for the purposes of updating the LRS with outcomes values of knowledge and skill attributes for all concepts and sub-concepts scored. The learner is then given the option to logout of the system, or to select a new course or lesson to complete.

3 Conclusion

In this paper we presented a use case of a conceptual course flow for a GIFT lesson managed by the EMAP. We highlighted architectural dependencies associated with building out an EMAP lesson and we reviewed logic associated with lesson transitions. This paper highlights the EMAP's function at the lesson level, where you can see the various decisions being made and the type of data informing its strategy selection. Enhancements to the EMAP continue, with current developmental plans looking at personalized feedback delivery options. In addition, the authoring process is being converted to web-based interfaces. For an overview of the current authoring process and to see the underlying features of the tools and methods put in place to support a pedagogical model like the EMAP, see [5] for a nice breakdown.

References

1. Poeppelman, T., Hruska, M., Long, R., Amburn, C. *Interoperable Performance Assessment for Individuals and Teams Using Experience API.* Paper presented at the Second Annual GIFT Users Symposium, Orlando, FL. 2015.
2. Shute, V. J., Kim, Y. J. Formative and stealth assessment *Handbook of research on educational communications and technology* (pp. 311-321): Springer. 2014.
3. Wang-Costello, J., Goldberg, B., Tarr, R. W., Cintron, L. M., Jiang, H. *Creating an Advanced Pedagogical Model to Improve Intelligent Tutoring Technologies.* Paper presented at the The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC). Orlando, FL. 2013.
4. Goldberg, B., Brawner, K. W., Sottilare, R., Tarr, R., Billings, D. R., Malone, N. *Use of Evidence-based Strategies to Enhance the Extensibility of Adaptive Tutoring Technologies.* Paper presented at the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2012, Orlando, FL. 2012.
5. Goldberg, B., Hoffman, M., Tarr, R. Authoring Instructional Management Logic in GIFT Using the Engine for Management of Adaptive Pedagogy (EMAP). In R. Sottilare, A. Graesser, X. Hu & K. Brawner (Eds.), *Design Recommendations for Intelligent Tutoring Systems: Authoring Tools (Volume 3)*: U.S. Army Research Laboratory. 2015.
6. Mitchell, J. L., Farha, N. Learning Object Metadata: Use and Discovery. In K. Harman & A. Koohang (Eds.), *Learning Objects: Standards, Metadata, Repositories, and LCMS* (pp. 1-40). Santa Rosa, CA: Informing Science Press. 2007.

Using Social Media with GIFT to Crowd-source and Enhance Learning Content

Irene T. Boland, PhD[1], Rodney A. Long[2], Ben Farmer, PhD[3], Doug Raum[1], Dan Silverglate[1], Ed Sims, PhD[1]

[1]Vcom3D, Inc., 12124 High Tech Ave., Suite 150, Orlando, FL, USA 32817
[2]Army Research Laboratory (ARL), Human Research and Engineering Directorate (HRED), Simulation and Training Technology Center (STTC), 12423 Research Parkway, Orlando, FL 32826
[3]Defense Equal Opportunity Management Institute (DEOMI). Patrick AFB, Florida
{ireneb, dougr, dans, eds}@vcom3d.com
rodney.a.long3.civ@mail.mil
benjamin.farmer.ctr@us.af.mil

Abstract. The US Army recognizes that enterprises that excel at incorporating their latest learning into the mainstream processes of their operations are able to capture and maintain a competitive edge. Among the goals of the Army Learning Concept 2015 is enabling all soldiers to participate in the creation and updating of training without increasing the workload of instructors. In addition to the Generalized Intelligent Framework for Tutoring (GIFT), the Army Research Laboratory (ARL) has funded a Social Media Framework (SMF) that enables an organization to crowd-source and crowd-vet new content and improvements to existing courses. The research questions we seek to answer in our current research include the extent to which the SMF and GIFT can: (a) promote critical thinking, collaboration, adaptability, effective communication, and problem solving; (b) help close the gap between formal training and operational application of the training to missions in the field; (c) reduce the time required to locate and use learning resources; (d) reduce the time required to incorporate feedback from the field into formal instruction; and (e) reduce instructor workload, while maximizing the efficacy of the instructor's time.

Keywords: Social media, GIFT, crowd-sourcing, usability, instructional systems design

1 Introduction

The US Army trains and educates over a half million individuals per year in a course-based, throughput-oriented system. Much of the Army's web-based instruction is in the form of static PowerPoint presentations, with little tailoring to individual soldier needs. With the ever-changing landscape of full spectrum operations, today's soldiers are facing ill-structured problems and have little time for the ideal levels of reflection and repetition needed to promote critical thinking, adaptability, and mastery of complex skills. Additionally, the current time frame for updating courses (3 to 5 years) is not supporting the modern Army's fast-paced learning needs.

During the Vcom3D demonstration of GIFT at the 17th International Conference on Artificial Intelligence in Education (AIED), attendees will experience how the breadth and depth of knowledge spread throughout an organization can be harnessed and rapidly incorporated into training for the benefit of those who need to know promptly. In the role of a learner, participants will experience and provide granular feedback on an adaptive course in our web-based GIFT environment. Then participants will discuss and vote on the relevance or accuracy of the content to enable refinement before an instructor reviews it for inclusion in training.

2 Background: Social Media Framework

Previously, we investigated a research-based suite of affordances that support the sharing and vetting of information amongst peers. The objectives of the project were to identify lessons learned from: commercial, academic, and US Government applications of social media to knowledge management and learning; and to consider the unique requirements and constraints of the military learning environment and how successful commercial and academic models for learning can be adapted to military applications.

3 Current Research

3.1 Research Objectives

At a high level, our research aims to investigate the extent to which the integrated SMF and GIFT system can:

- Promote critical thinking, collaboration, adaptability, effective communication, and problem solving,
- Help close the gap between formal training and operational application of the training to missions in the field,
- Reduce the time required to locate and use learning resources,
- Reduce the time required to incorporate feedback from the field into formal instruction,
- Reduce instructor workload, while maximizing the efficacy of the instructor's time.

3.2 Experimental Methodology

This research project follows a sequence of overlapping/spiral events, including: Literature Review (ensuring that our proposed research furthers the body of knowledge), Experiential Review (hands-on examination of existing, to ensure that the affordances we test are extending the state of the art), Test Bed Development (creating the suite of affordances to enable testing of our research hypotheses), and Quantitative and Qualitative Research (testing our hypotheses and soliciting feedback from participants).

3.3 Test Bed Architecture

Expanding on the existing SMF, a cloud-based 'headless' instance of the GIFT platform has been created, allowing multiple users to connect to GIFT across the internet.

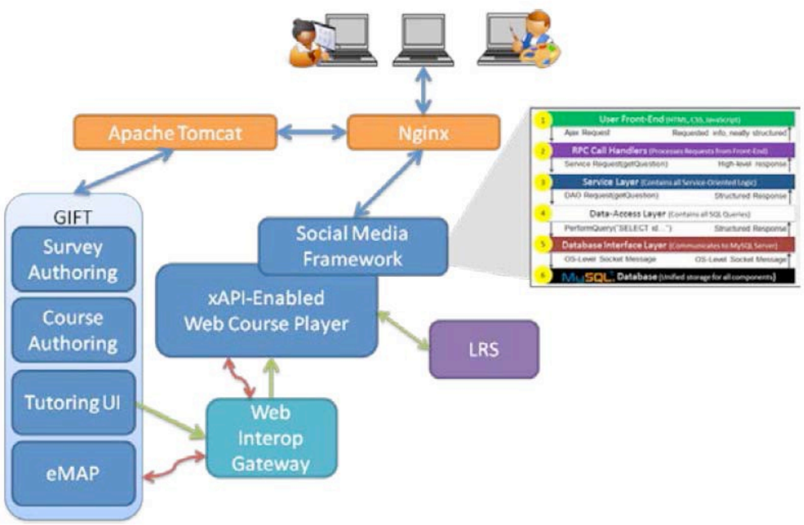

Fig. 1. SMF/GIFT Integrated Architecture

The GIFT platform has been extended to include a gateway interoperability module allowing for connection to a web-based course player. The course player, built on a PHP/MySQL platform and using a responsive front-end (suitable for expansion to mobile devices), will play (experience API) xAPI-wrapped course content. Through the gateway interoperability module, the course player will also communicate to the GIFT engine for Management of Adaptive Pedagogy (eMAP), allowing adaptivity within the course driven by GIFT's advanced adaptive capabilities. The course player also generates xAPI statements which are stored in a Learning Record Store (LRS) and usable for learning analytics.

The web-based course player includes the ability for courses to collect social media feedback on granular aspects of the course: paragraphs of text, images, videos, etc.

Using annotation-style commenting, the social feedback is collected and stored within the SMF for crowd-comment and review after the course is completed. In addition, the GIFT tutoring user interface (UI) has been modified to allow other GIFT transitions (surveys, learning materials, after action reviews) to collect social feedback in a similar manner. This feedback, too, will be available within the SMF for crowd-comment and interaction.

3.4 Experimental Research

Vcom3D research for the ARL in Social Media-enabled Learning and Knowledge Management has three major phases in 2014-2015, each with a data collection. The recently completed (February 2015) data collection 1 focused on having an Instructional Systems Designer (ISD) and SMEs use a Learning Content Management System (LCMS) to enter content and build a course. The research test bed is a combination of three government-sponsored systems: SMF, GIFT, and an LCMS.

The second data collection (Summer 2015) will involve learners taking the course and providing granular feedback about how they think the course can be improved as well as using social media tools to discuss the feedback of others. Then, in data collection three (Fall 2015), the ISD and SMEs who built the course will review the feedback from learners and decide what improvements they will make to the course. This three-part research demonstrates the speed with which experts in the field and fleet can provide real-world feedback that is then promptly incorporated into the official doctrine course by the schoolhouse. This addresses key goals of the Army Learning Model (ALM) which seeks, among many other goals, to include the ever-evolving knowledge of the field and fleet into the official training as quickly as possible.

Data Collection 1 Procedure. Expanding on the existing SMF, a 'headless' instance of ARL's GIFT platform was created, allowing it to run independently of a specific workstation. Utilizing this, we deployed the GIFT Survey Authoring System (SAS) and GIFT (CAT) Course Authoring Tools through our existing Apache Tomcat web application server. Using nginx to serve the existing SMF and act as a proxy to the GIFT instance on the same server gave the participants the experience of a seamless, consolidated system with Single Sign On (SSO) for each subsystem. The experimental test bed was hosted on a dedicated server off site from the research location. Each participant received login credentials and used a separate work station in their lab to access the test bed though the internet from a standard browser.

The researchers guided participants through standard tasks involved in creating learning content. The session was videotaped to allow for detailed analysis afterward. We described the system to our participants as an experimental learning content authoring system the Army has asked us to build and test. We explained that our long-term goal is to grow the system into a powerful tool that is useful to them (and other users) in creating adaptive learning experiences that are easy to update. Having their formative feedback at this early stage will enable us to develop it in the direction that's most useful to users.

We designed their data collection experience to simulate a collaboration to create the course. So, each participant was asked to create a different scenario and then we had them work together to tie it all into a complete course.

Data Collection 1 Results. Each of our recommendations has its basis in the time-tested and research-proven principles of UI and User Experience (UX) professions. Our recommendations are meant to help move GIFT closer to its goal of being useful to SMEs who want to author effective courses on their own. The Nielsen/Norman Group of UI/UX professionals defines useful as the result of usability and utility. Utility speaks to the extent that the system has the features the user wants and needs. Usability can be described as having 5 criteria: (1) easy to learn to use, (2) user can complete tasks quickly, (3) user can remember how to use it after being away from it for a while, (4) errors the user makes are few and easily rectified, and (5) the system is enjoyable to use.

Recommendation 1: Sell the utility, immediately. Users found that the system contained a large number of steps compared to other systems they had used to build adaptive training or surveys. Some of those steps were unclear in meaning or purpose. The naming conventions used are not consistent with what SMEs would name the features, buttons, and other controls. As a result, they expended a great deal of mental effort (cognitive tolls) to work in the system. Although the researchers explained the long term purpose of the system (to creative adaptive training suited to each individual), the perceived benefits of the system were not sufficient to motivate the users to want to continue using the system in its current state. For all of these reasons, we recommend an early intervention of Selling the utility – making the benefits of the system so clear that new users will be motivated to expend the needed effort to understand and master the system.

We recommend the system provide a short but impactful explainer video that helps users understand the system and what's in it for them. Specific questions that should be answered include: (a) What is Adaptive Learning? (b) Why should I use Adaptive Learning with my learners? (c) What is GIFT? And, why is it better than my other options? (d) How have others similar to me used it (compelling real success stories/visuals)? and (e) How do I use GIFT to create Adaptive Learning?

The military has a long-standing tradition of rigorous ISD which follows a standard ADDIE model (analysis, design, development, implementation, evaluation) of activities. We can reasonably expect a SME to have extensive knowledge of the content being taught. Based on their experience, they may also bring knowledge of the audience (having been a trainer) and the related organizational goals that lead to the SME being asked to share their knowledge. However, there are significant knowledge gaps in ISD for most SMEs. To achieve the long term goal of an independent SME creating effective training, the system must provide the education and support needed by the SME.

Recommendation 2: Use the process and vocabulary native to the SME. The current process flow and vocabulary used in the system is not reflective of how most SMEs think or work. As a result, they are burning significant brain power simply trying to understand the system rather than feeling the reinforcement of accomplishing their goals. To illustrate both of these concepts, we examined a short process – Adding a question to an assessment – as SMEs are accustomed to doing it compared to how SMEs attempt to do it in GIFT.

For this very short sub-process of the larger course creation process, we can compare the expected versus experienced using the scorecard shown in Table 1.

Table 1. Cognitive Load Comparison

Measure	*GIFT Experience*	*Usual SME Experience*
Steps	20 (steps 7-9 repeat 3X)	9 or less*
Cognitive Load	High	Low
Time	Slow	Medium
Other	Process incomplete. Feedback to be added using additional steps, time and cognitive load in another part of GIFT.	* Ability to upload can make process even shorter.

Recommendation 3: Incorporate extensive, yet lean, on-demand contextual support for SME. We recommend two approaches to providing support to SMEs. First, provide them some fast and simple support when they first arrive. This help should display automatically the first time the user experiences a screen. Afterward, it should be available for the user to display on demand).

Second, offer mouseover-based help for each control, vocabulary term or other element that the SME might not be familiar with. The example in Figure 2 shows that a vocabulary improvement has been made – changing the word Transition to Content, and then providing a mouseover that explains what particular types of content are and alerting the user if they will need to use another part of the system to create that content before trying to use it here.

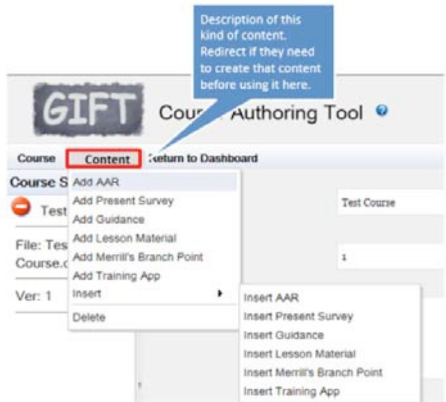

Fig. 2. Mouseover Help Example

Data Collection 2 Procedure. The SMF will be expanded to include course topics and the actual courses in the training tab. Once launched, courses will be played through the GIFT framework. In GIFT, a course is a series of transitions which might include Surveys, Learning Materials, and Training Applications. To enable a Training Application to play lessons comprised of web-based content, we will implement a new gateway interoperability module. Unlike standard web-based lessons, however, any element of the content can be selected and commented upon. Showing those comments in close proximity to the lesson content could negatively impact the flow of the course for future learners; so instead, the comments will automatically appear as a new conversation thread under the feedback tab of the containing topic page for this course. We will add similar social media commenting capability to other GIFT transitions such as Surveys and Learning Materials. The course material will be furnished by DEOMI ISDs and will be selected for its relevance to the target student participants for specific use in the experimental research. The content will then be prepared for playback by the web-based lesson Training Application and other GIFT transitions.

During the data collection event, multiple sessions of approximately 20 student participants each will access the experimental test bed from work stations in their lab through the internet from a standard browser and using credentials provided by the researchers. Participants will be asked to navigate to a particular topic and take the course associated with that topic. Participants will be encouraged to generate questions or feedback on any content they encounter. After completion of the course, participants can review their comments on the topic page and also see the comments of other participants. They will be able to up vote and down vote the questions, answers, and feedback generated by others as well as contribute to the discussions. Participants in subsequent sessions will the accumulated contributions of all preceding participants. At the end of each session, the participants will complete a survey to provide feedback of their experience.

Data Collection 3 Procedure. The third phase of research will explore techniques and algorithms for analyzing the user-created content, surfacing the most relevant comments and activity and connecting them to the most relevant stakeholder. For this data collection with content authors and content owners, the user management section of the SMF will evolve to display a user digest specific to each user and their role in the system. An activity section will highlight the latest contributions by the user. Back-end data analytics will look at factors such as up votes, down votes, and general activity to prioritize the contributions of others relevant to this user. The goal is to highlight trending and actionable issues pertaining to course content owned by this user. Participants will then evaluate the efficacy of the system in surfacing errors, identifying gaps, suggesting content, and reducing ISD work-load.

4 Implications for Future Research

At the end of the third phase of the current research, we will have investigated the efficacy of crowd-sourced and crowd-vetted content for applying field knowledge to improve learning content, while reducing instructor workload and turn-around time. However, we believe that social media can provide additional benefits to the learning environment, and to GIFT in particular, by (1) harnessing crowd inputs for the creation and refinement of a Domain Model, or the body of knowledge for a topic and (2) mining social media data to enhance an individual's Learner Profile (or personal history of learning, demographics, and achievements). We have also identified the need to make the user experience more intuitive to its intended end-users (SMEs). At the end of the current research, we will make recommendations for these additional means for applying social media to the integrated learning environment.

Additional areas of research we intend to explore include: (1) harnessing crowd inputs into the creation and refinement of a domain model, or the body of knowledge for a topic, (2) mining social media data to enhance an individual's Learner Profile (or personal history of learning, demographics, and achievements), and (3) developing the user experience to be immediately intuitive to its intended end-users (fielded subject matter experts).

References

1. Advanced Distributed Learning (ADL). Experience API: Research Summary. Retrieved May 13, 2013 from http://www.adlnet.gov/tla/experience-api
2. Nielsen, J. Usability 101: Introduction to Usability Available. 2012: http://www.nngroup.com/articles/usability-101-introduction-to-usability/
3. US Army TRADOC (2011). The US Army Learning Concept for 2015. Retrieved November 15, 2012 from http://www.tradoc.army.mil/tpubs/pams/tp525-8-2.pdf

NewtonianTalk: Integration of Physics Playground and AutoTutor using GIFT

Matthew Ventura[1], Xiangen Hu[2,3], Benjamin D. Nye[2], Weinan Zhao[4]

[1]Empirical Games, Tallahassee, FL 32312
[2]University of Memphis, Memphis, TN 38152
[3]Central China Normal University, Wuhan, China. P.R.C.
[4]Florida State University, Tallahassee, FL, 32312
matthewventura@empiricalgames.org; {xhu,bdnye}@memphis.edu;
weinan.zhao@gmail.com

Abstract. Despite the popularity of games, there has been limited peerreviewed literature published on game-based learning for science. This paper will describe a project that combined an Intelligent Tutoring System (AutoTutor) with a physics game called Physics Playground. As part of this integration we used the Generalized Intelligent Framework for Tutoring (GIFT) to manage communication between the two technologies. We will also discuss the design of a study comparing two versions of the integration. This study is taking place over Spring of 2015 and will be studying the effects of integrating different levels of tutoring into a gamebased learning system.

Keywords: Game-based Learning, Intelligent Tutoring Systems, Physics, Playground, AutoTutor, GIFT

1 Introduction

There is growing evidence of video games supporting learning (e.g., Tobias & Fletcher, 2011; Wilson et al., 2009). Such research typically focuses on games explicitly designed for learning. However, games not explicitly designed for learning can also produce significant learning gains. In this research, we look at the potential benefits of adding intelligent tutoring into an existing game. This paper describes the design process for creating an ITS enhanced educational game called NewtonianTalk using the GIFT technology. Before we describe the integration we will briefly review the state of ITS and educational games.

2 Background

2.1 Intelligent Tutoring Systems

Intelligent tutoring systems (ITS) have proven very effective in improving training outcomes. Meta-analyses show effect sizes on the order of one sigma (Dodds &

Fletcher, 2004; VanLehn, 2011), which is approximately a full letter grade in traditional grading schemes. The long sought-after goal is a 2σ effect size (Bloom, 1984; Corbett, 2001).

Recent advances in natural language processing (NLP), semantic analysis, machine learning, and cognitive modeling have spawned ITSs with the potential to achieve this effect size (Graesser, Conley, & Olney, 2012). Although many of the current computer tutors tend to use heuristics that remain constant as they customize material for individual students, the next generation of tutors will implement more dynamic models that can infer hidden learner characteristics and recognize unanticipated behavior based on learner performance, past experiences, and lessons learned. Aside from these breakthroughs in AI, the next-generation ITSs may include game-like components that further engage the student in the learning experience.

In the research discussed here, the AutoTutor Lite ITS (ATL, Hu et al., 2009) uses an established method of engaging a learner in a natural-language tutorial dialog (Graesser, Olney, Haynes & Chipman, 2005). ATL appears as an animated "talking head" avatar at certain points during the game and engages the learner in conversation about key physics concepts.

2.2 Learning Support via Games

Well-designed games can be seen as vehicles for exposing players to intellectual problem solving activities (Gee, 2004). But problem solving can be frustrating, causing some learners to abandon their practice and, hence, learning. This is where the principles of game design come in: Good games can provide an engaging and authentic environment designed to keep practice meaningful and personally relevant. With simulated visualization, authentic problem solving, and instant feedback, computer games can afford a realistic framework for experimentation and situated understanding, and thus act as rich primers for active learning (Shute & Ventura, 2013).

Furthermore, within-game learning support enables learners to do more advanced activities and to engage in more advanced thinking than they could without such help. The complicated part about including learning support in games is providing support that does not disrupt engagement while learners are immersed in gameplay, and reinforcing the emerging concepts and principles that deepen learning and support transfer to other contexts.

2.3 Physics Playground

Research into what is called "folk" physics demonstrates that many people hold erroneous views about basic physical principles that govern the motions of objects in the world, a world in which people act and behave quite successfully (Reiner, Proffit, & Salthouse, 2005). Recognition of the problem has led to interest in the mechanisms by which physics students make the transition from folk physics to more formal physics understanding (diSessa, 1982) and to the possibility of using video games to assist in learning (Masson, Bub, & Lalonde, 2011).

The game Physics Playground (PP) was designed to help middle school students understand qualitative physics (Ploetzner, & VanLehn, 1997). We define qualitative physics as a nonverbal understanding of Newton's three laws, balance, mass, conservation of momentum, kinetic energy, and gravity. PP is a 2D sandbox game that requires the player to guide a green ball to a red balloon. The player can nudge the ball to the left and right (if the surface is flat) but the primary way to move the ball is by drawing/creating simple machines on the screen that "come to life" once the object is drawn. Everything obeys the basic rules of physics relating to gravity and Newton's three laws of motion. Using the mouse, players draw colored objects on the screen, which "come to life" as physical objects when the mouse button is released. These objects interact with the game environment according to Newtonian mechanics and can be used to move the ball. When objects interact within the game environment, they act as "agents of force" to move the ball around. The player creates simple levers, pendulums, and springboards to move the ball.

The difficulty of a puzzle was based on a number of factors including: relative location of ball to balloon, number of obstacles present, number of agents required to solve the problem, and novelty of the problem. Difficult problems provide greater weight of evidence to the estimate of a competency level than easy problems. Also, "elegant" solutions (i.e., those using a minimal number of objects) give greater weight to competency level inferences than regular solutions. Preliminary data suggest playing PP for four hours can improve qualitative physics understanding ($t(154) = 2.12$, $p < .05$) with no content instruction or other learning support (Shute, Ventura, & Kim, 2013).

3 Methodology: GIFT Management of ATL and PP

As education turns to more game-like ITS learning environments it is important to ensure that their learning pedagogy remain consistent with the learning sciences. To ensure a good balance between the motivating "skin" of the learning experience and the deep "muscle and skeleton" of science-based learning, it is important to adopt a general architecture of ITS learning. The GIFT framework provides such an architecture and allows the integration of independent learning technologies (Graesser, Hu, Nye & Sottilare, In Press). In this work, GIFT manages and controls data communication between ATL and PP.

While the vast majority of the components of an ITS may be made domain independent, there must always be a specific component of the architecture to deal with the problems that the instructor desires to teach. The fundamental problems of domain-dependent components are how to assess student actions, how to respond to instructional changes, how to respond to requests for immediate feedback, and an interface that supports learning (Sottilare, Goldberg, Brawner and Holden, 2012; Goldberg, Sottilare, Brawner, & Holden, 2012). The architecture designed must have built-in support for these types of instructional activities.

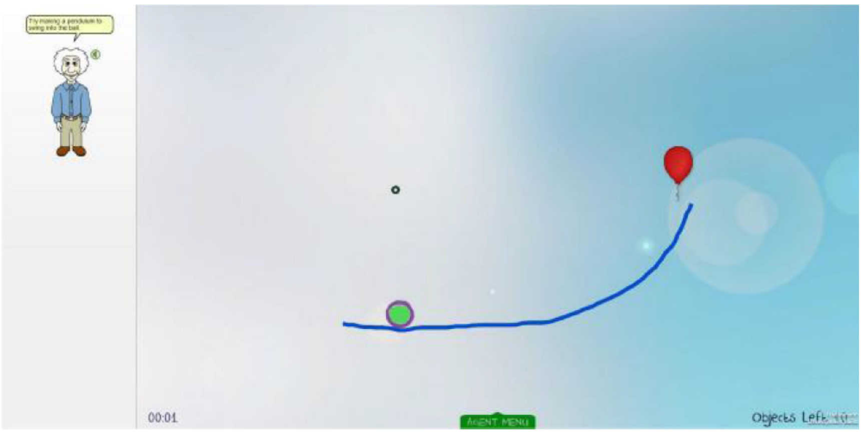

Fig. 1. NewtonianTalk Interface

Figure 1 displays the interface of NewtonianTalk. As can be ATL is always displayed on the left next to the PP interface. There are 3 playgrounds in NewtonianTalk. Each playground teaches a physics concept with 3 puzzles (Impulse, Conservation of Momentum, Conservation of Energy). The first design decision that needed to be made was how to most effectively introduce dialogue into PP without disrupting game play. We chose the following pedagogy styles for instruction: information delivery through ATL, scaffolded question and answer selfexplanation in ATL, and PP puzzles with support instruction. The selection of the specific activity is handled by rules specified in the GIFT system that act conditionally on information sent from the PP puzzle as the student interacts with it. Below is the introductory explanation of Impulse to the player:

> *An unbalanced force can cause an object to speed up or slow down. Specifically, an impulse is required to change the speed of an object. Impulse is the product of force times time. To change ball's speed, a springboard exerts a force for an amount of time. Pulling the springboard down further increases the ball's speed even more by applying a greater force for a longer time.*

After the player listens to further explanation as they play three PP puzzles. Figure 2 displays the puzzle for Impulse. As the springboard exerts a force up on the ball for an amount of time, it gives an impulse to the ball that changes the ball's motion. Increasing springboard's force or the time the springboard pushes up on the ball causes it to go even higher.

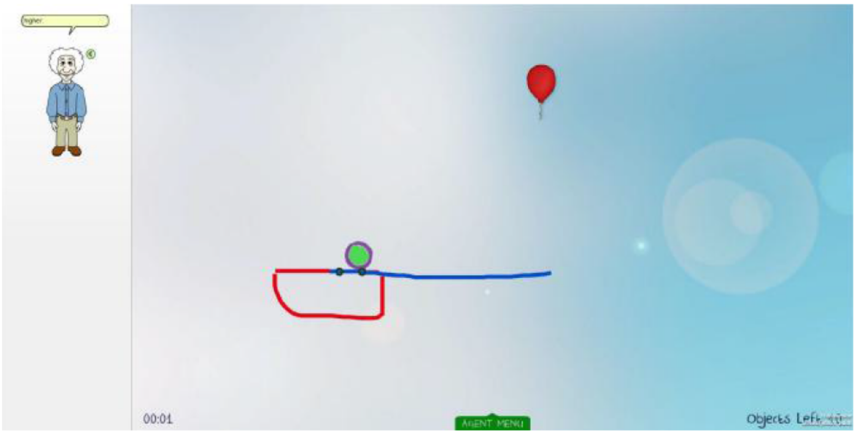

Fig. 2. Impulse puzzle with explanatory audio

After the player solves all 3 puzzles in the playground ATL poses a series of questions in natural language. Automated scores are calculated for the learner's performance. Below are questions for impulse:

Q. What is impulse? **A.** Impulse is force times time.
Q. How does an impulse affect an object? **A.** An impulse can change an object's speed.
Q. How could a force make a larger impulse? **A.** Increase the force or increase the amount of time.
Q. How can the same impulse be applied if the time of contact is reduced? **A.** To apply the same impulse over a smaller amount of time, the force must increase.

Once the player has answered the questions correctly or has maxed out the attempts (3 per question), the player then moves to the next playground. The player is given feedback in terms of percentages of completing the playgrounds and the ATL questions.

4 Discussion and Future Directions

This design process for this integration has identified some of the strengths and challenges for adding intelligent tutoring to an existing game environment that is mainly focused on simulation and experimentation. A strength of adding ITS interactions to such a game is that it allows instruction and discussion of the principles involved as they are encountered in the game (or, alternatively, fill them in when the learner struggles). Prior research on learning through exploring simulations indicates that such help may be important to learn from these activities efficiently (Graesser, Chipman, Haynes and Olney, 2005).

This approach can also be used as a model to enhance noneducational games to make them more effective for learning. For example, the game Portal 2 (despite not being learning-focused) showed significant benefits for certain types of problem solving skills (Shute, Ventura, and Ke, 2015). The current research integrates ITS into a Unity game, which is a popular engine. Such games may prove powerful learning environments with intelligent tutoring used to highlight and connect the key principles and concepts. However, the primary challenge of this work is to be able to integrate tutoring into an existing interface without being disruptive or introducing too much cognitive load.

We will be collecting data on NewtonianTalk in 2015 on an estimated 100 undergraduate psychology students. In addition to getting valuable usability data we also will test a hypothesis regarding instruction pedagogy. For this study, additional functionality is being specified that will leverage the ability of GIFT to manage and coordinate just-intime feedback based on the learner's activities during a playground. Learners' freedom to explore in a playground may increase transferability of skills, but may also result in unproductive exploration. It is hoped that GIFT support will make exploration more effective.

Acknowledgments. This work was supported by the Army Research Lab grant W911NF-12-2-0030 and applies the open-source GIFT system as developed by the ARL Learning in Intelligent Tutoring Environments (ARL-LITE) lab and Dignitas. However, the views of this paper represent only those of the authors.

References

1. diSessa, A. A. (1982). Unlearning Aristotelian physics: A study of knowledgebased learning. Cognitive Science, 6, 37–75
2. Dodds, P., & Fletcher, J. D. (2004). Opportunities for new" smart" learning environments *enabled by next generation Web capabilities* (No. IDA-D-2952). Institute for Defense Analyses: Alexandria, VA..
3. Gee, J. P. (2004). What Video Games Have to Teach Us about Learning and Literacy. Palgrave Macmillan.
4. Graesser, A. C., Chipman, P., Haynes, B. C., & Olney, A. (2005). AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. *Education, IEEE Transactions on*, *48*(4), 612-618.
5. Graesser, A. C., Conley, M. W., & Olney, A. (2012). Intelligent tutoring systems. In K.R. Harris, S. Graham, T. Urdan, A. G. Bus, S. Major, H. L. Swanson (Eds.) *APA educational psychology handbook, Vol 3: Application to learning and teaching*.(pp. 451-473). Washington, DC, US: APA.
6. Graesser, A.C., Hu, X., Nye, B. & Sottilare, R. (In Press). Intelligent Tutoring Systems, Serious Games and the Generalized Intelligent Framework for Tutoring. In H.F. O'Neil, E.L. Baker & R.S. Perez. (Eds.), *Using games and simulation for teaching and assessment*. Routledge: Abingdon, Oxon, UK.
7. Goldberg, B., Sottilare, R., Brawner, K., & Holden, H. (2012). Adaptive Game- Based Tutoring: Mechanisms for Real-Time Feedback and Adaptation. In *International Defense & Homeland Security Simulation Workshop in Proceedings of the I3M Conference*. Vienna, Austria, September 2012.
8. Hu, X., Cai, Z., Han, L., Craig, S. D., Wang, T., & Graesser, A. C. (2009). AutoTutor Lite. In Artificial Intelligence in Education (*AIED) 2009* (pp. 802-802). IOS.

9. Masson, M. E. J., Bub, D. N., & Lalonde, C. E. (2011). Video-game training and naive reasoning about object motion. Applied Cognitive Psychology, 25, 166–173.
10. Reiner, C., Proffitt, D. R., & Salthouse, T. (2005). A psychometric approach to intuitive physics. *Psychonomic Bulletin and Review, 12*, 740–745.
11. Ploetzner, R., & VanLehn, K. (1997). The acquisition of qualitative physics knowledge during textbook-based physics training. *Cognition and Instruction, 15*(2), 169-205.
12. Shute, V. J., & Ventura, M. (2013). *Measuring and supporting learning in games: Stealth assessment*. Cambridge, MA: The MIT Press
13. 13. Shute, V. J., Ventura, M., & Ke, F. (2015). The power of play: The effects of Portal 2 and Lumosity on cognitive and noncognitive skills. *Computers & Education*, *80*, 58-67.
14. Shute, V. J., Ventura, M., & Kim, Y. J. (2013). Assessment and learning of qualitative physics in newton's playground. *The Journal of Educational Research*, *106*(6), 423-430.
15. Sottilare, R. A., Goldberg, B. S., Brawner, K. W., & Holden, H. K. (2012, December). A modular framework to support the authoring and assessment of adaptive computer-based tutoring systems (CBTS). In *Interservice/Industry Training, Simulation, and Education Conference.(I/ITSEC) 2012*. Paper 12017 (pp. 1-13).
16. Tobias, S., & Fletcher, J. D. (Eds.). (2011). *Computer games and instruction*. IAP.
17. VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, *46*(4), 197-221.
18. Wilson, K. A., Bedwell, W. L., Lazzara, E. H., Salas, E., Burke, C. S., Estock, J. L.,... & Conkey, C. (2009). Relationships between game attributes and learning outcomes review and research proposals. *Simulation & Gaming*, *40*(2), 217-266.

Rapid Dialogue and Branching Tutors

Keith Brawner[1]

[1]U.S. Army Research Laboratory, Orlando, FL 32826
Keith.w.brawner.civ@mail.mil

Abstract. The technology used as part of the Tools for Rapid Automated Development of Expert Models (TRADEM) project has been featured at a number of conferences and publications throughout its creation and development. As a part of these efforts, it has been integrated with the Generalized Intelligent Framework for Tutoring (GIFT) in two fashions: branching, using the Engine for Management of Adaptive Pedagogy (EMAP), and dialogue-based, using open-source chat technology. This technology is nearly ready to be deployed to the public, enabling this workshop to demonstrate its capability, highlight its use, and allow users to make their own tutors centered about their own content.

Keywords: intelligent tutoring system, ADDIE process, dialogue based tutoring, branching tutoring

1 Introduction

The Tools for Rapid Automated Development of Expert Models (TRADEM) project was first published in 2013 in a simulation venue [1]. The technology was demonstrated last year at the Intelligent Tutoring Systems 2014 conference, as part of a workshop on authoring tools [2], at the Educational Data Mining 2014 conference, as part of an industry session [3], and at the annual GIFT Symposium, as part of general GIFT development [4]. The project has recently come to completion, with the outputs intended to be made publicly available soon, and physically distributed as part of this workshop.

As described by many, including the GIFT foundation paper [5], Intelligent Tutoring Systems (ITSs) contain four components: a domain model, an expert model, a learner (or student) model and a pedagogical model. TRADEM uses a domain model built as a summarization of provided content mixed into a set of topics, as a part of the GIFT Domain Module. The expert model consists of a domain model together with expert-derived information concerning the order of topic learning, information about the content, and a basic manner of assessing learner response. These pieces of information are represented in the GIFT Domain Knowledge File (DKF), and are linked with a series of questions in the Survey Authoring System (SAS). The pedagogical model used as part of TRADEM-produced tutors is simply the GIFT default engine, called the Engine for Management of Adaptive Pedagogy (EMAP), which has been documented in greater detail in other literature [6].

The purpose of the TRADEM project has been to rapidly and mostly-automatically create expert models and sequence domain material from initially provided texts. The traditional teaching model relies upon teachers to select the material for consumption by the learners, where the teacher provides the material. The TRADEM model of development is to condense the material selected for students, where the system provides the learning material created from previously provided learning materials. Naturally, there is some disagreement in the literature as to the nature of an "expert model."; is it the selected materials by the teacher, or the core concepts identified by the system? In the TRADEM formulation, a domain model consists of a set of topics in a domain, while an expert model consists of a domain model together with expert-derived information concerning the order in which topics should be learned and expert-derived data that enables an ITS to present each topic and assess learner knowledge. Expert derived information may take a few different forms. The first of these are the topic names and conventions used as a map of the topics, as shown later in Fig. 3 and Fig. 6. The second part of the expert-derived information is in metadata about the type of information content contains (e.g. Gagne's 9 Events [7] or Merrill's Component Display Theory [8]). The last of the expert-derived information is questions and answers, which are automatically suggested based on the content, and curated by the human expert.

This paper is intended to briefly describe the how the system operates and the technologies which it relies upon, as a short description is helpful to the reader, although not required for practical use. In practice, the purpose of the workshop of this technology is to demonstrate the technology. In short, TRADEM uses automated text analysis techniques to create core groups of "topics" based upon the topics that appear to have been discussed the most. It uses automated summarization techniques to create summary text paragraphs and link it to an exact topic, and uses this text to propose a name for the topic, as content for the topic, and as a basis for creating questions. The technical tasks to perform each of these items are described in other works throughout the literature [1-4].

2 Use

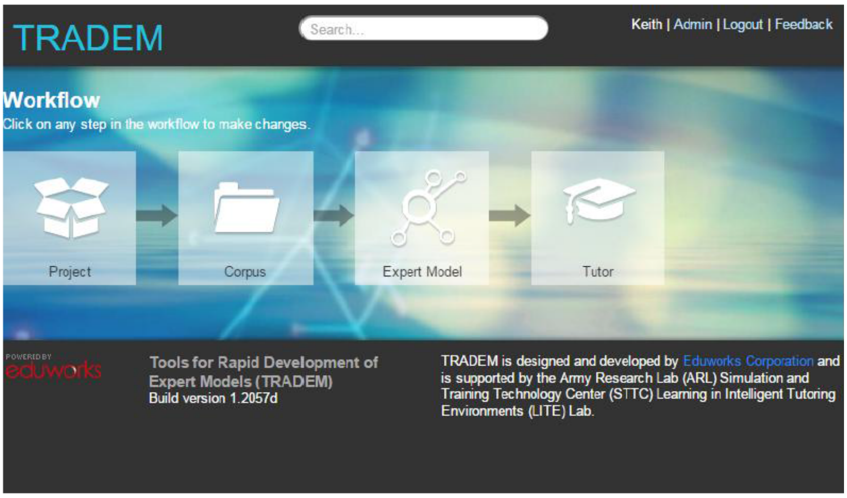

Fig. 1. TRADEM User Interface

The basic process of creating a tutor with TRADEM is simple, and relies upon a few basic steps, all of which are shown from the screen following login, as seen in Figure 1. In this section, we will highlight the specific steps required to produce a tutor within the TRADEM authoring workflow.

Step One: Create a new project and give it a name.
Step Two: Create a corpus, upload documents to it, and save, as seen in Figure 2.

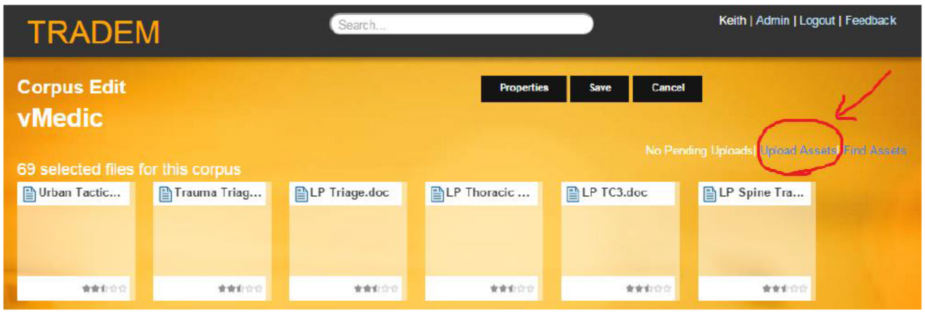

Fig. 2. Corpus creation and editing

Step Three: Add a new expert model through a selection of features. TRADEM provides an estimation of the number of topics present within your model when using the default settings. If your corpus has a fewer number of documents, or some of the documents in your corpus are short but contain critical information, you may consider adjusting the expert model parameters to be higher than the default values, shown in Figure 3.

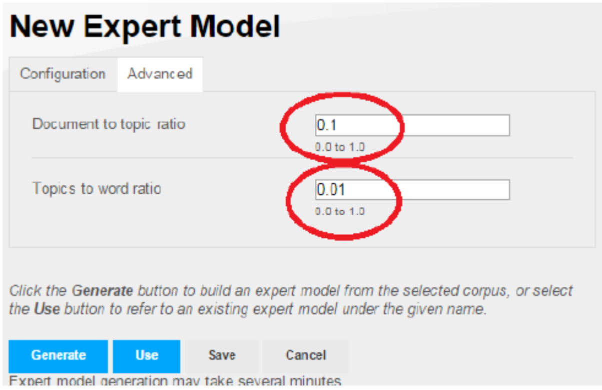

Fig. 3. TRADEM Expert Model Parameters

Step Four: Edit the expert model and mini-corpus. Be sure to have enough questions on each topic to support the GIFT default exports (3 questions per topic). If TRADEM has not suggested enough questions related to the topic, the user may have to create them manually or generate a new expert model. See the highlighted area in Figure 4 to edit the topic in this manner.

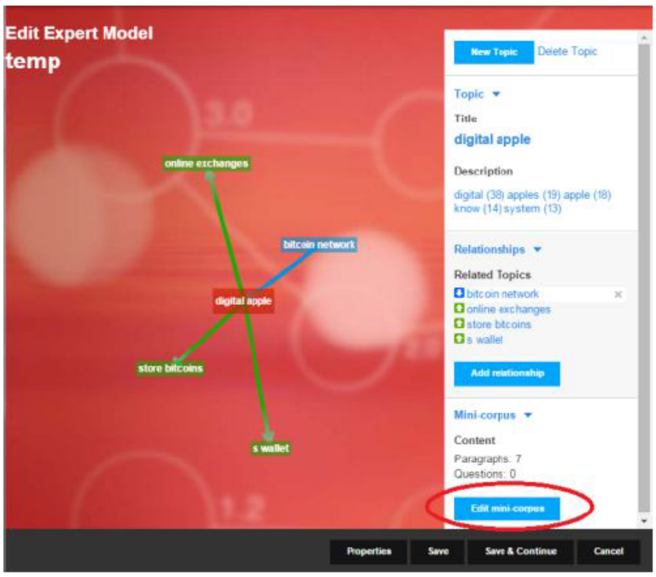

Fig. 4. Expert Model Editing

Step Five: Export the tutor. At this point you will receive three options to either 1) export as a standard package, 2) export as a GIFT TRADEM-Tutor ("T-Tutor") pack-age, or 3) export as a GIFT PowerPoint (PPT) package. The first of these options exports unadorned slides and questions/answers for presumed import into other Learning Management Systems (LMSs) and traditional training content. The second option exports a dialogue-based "talking head" which can understand basic student inputs and course directions, and can be imported into GIFT. The third of these options exports a series of PowerPoint shows and pre-/post-tests which can be imported into GIFT and managed as a branching course. These options are shown in Figure 5 which shows the "export tutor" option and the "generate export" option after selecting one of the above three choices.

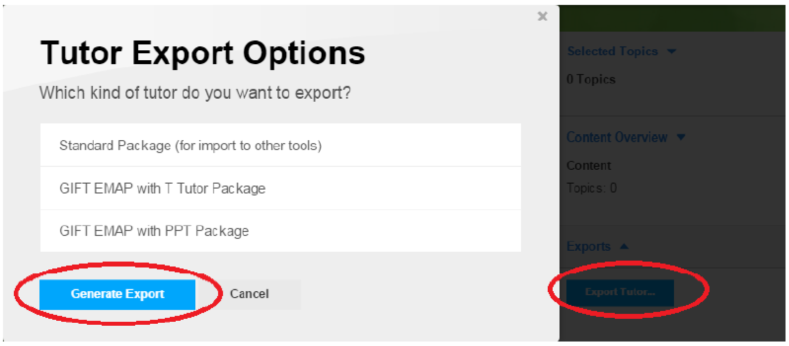

Fig. 5. TRADEM Export Tutor Dialogues

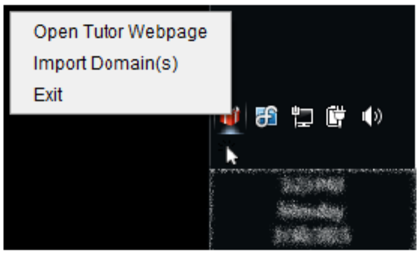

Fig. 6. GIFT Import

Step Six: Import the package into an existing GIFT installation using the GIFT Import Tool. The GIFT import tool can be found by right-clicking on the GIFT icon as shown in Figure 6, or in the GIFT\scripts\tools\launchControlPanel.bat interface. After import, the EMAP course will be selectable and display as traditional PowerPoint slides, while the "TTutor" export will display with a "talking" head and simplistic dialogue responses, as shown in Figure 7.

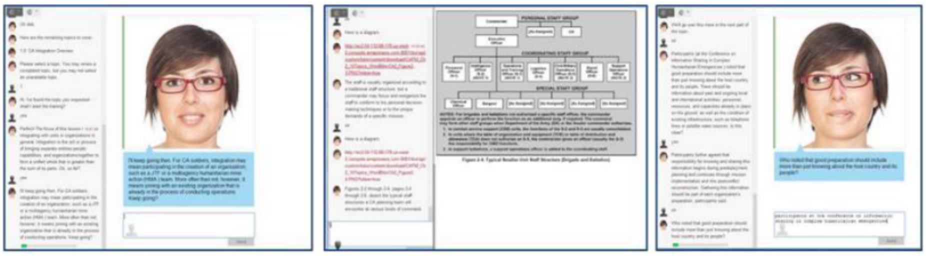

Fig. 7. TRADEM-Tutor Interface [4]

3 Benefits for Use

There are a few benefits to using the TRADEM tool, including aiding in front end analysis of content, automatically summarizing existing documents, or providing the foundation of a GIFT course. This section briefly discusses these three use cases.

One of the manners of TRADEM use is to perform a front end analysis of the content being worked with. The import of content into TRADEM and looking at the structure of the domain can prove valuable to deciding other methods of instruction. As an example, differing domains may represent different manners of instruction, as shown in Figure 8 with a few different domains. This analysis may affect human decisions of how to instruct the material, and can be garnered fairly quickly (minutes).

Fig. 8. Discovered organizational structures [3], which may be instructed differently

A second manner of technology use is in the automated summary of learning materials. The automated summarization techniques can be used with conference track papers as input, and presented a summary of the things discussed in the individual tracks [1]. Such use may be able to guide conference learners to the sessions of their greatest interest, based on the papers accepted to the tracks.

Further, a GIFT tutor which uses the EMAP can be created with very little effort through the use of TRADEM. Instead of uploading various learning materials, tagging them with metadata, and building a course, the TRADEM tool can be used to integrate checkboxes for metadata, and automatically sequence the content. Given the speed and simplicity of use, such practice may prove standard to the creation of GIFT-EMAP courses. This allows tutor creators to benefit from an extensively researched instructional domain model without significant investment of time, and using content which can be fine-tuned at a later time with the GIFT authoring tools. Other benefits are more extensively discussed in other works [3].

3.1 Licensing

The open-source nature of GIFT means that reproducible code is freely released and updated with each subsequent version. Tutors, the output of GIFT, are free to produce and may be sold or freely provided for community benefit. Developed modules and plug-ins may additionally be sold or donated, while GIFT components may never be sold. While TRADEM is free for both use and modification in Government applications, it is not open source. The close-source encumbrances of TRADEM, however, are not burdensome. The closed-source encumbrances are 1) that the user must agree to a licensing agreement on branding prior to the generation of tutoring materials, and 2) not to remove the branding of the tutoring materials created as part of the TRADEM process. Aside from these issues, the tutors produced using the TRADEM process are free to be used and commercialized as GIFT outputs.

4 Future Work

The primary use of TRADEM is for use as an advanced and automated authoring capability [9], but there is a follow-on effort to automate the process of evaluating the weaknesses of the produced courses. The intention is that an instructor, after creating a GIFT or TRADEM course, would be able to analyze the course for the items that produce (or omit) learning gains on the relevant post-test measures. Additional measures are being taken to change the login/logout credentials to match GIFT, to make the Gateway Module plug-in an interoperable and separable service, and to enable web-based learning and software testing. The current architecture and integration is shown in Figure 9, and represents a way for other dialogue tutoring services to integrate into GIFT, as they can either follow this example integration, or the one provided by the AutoTutor webservices.

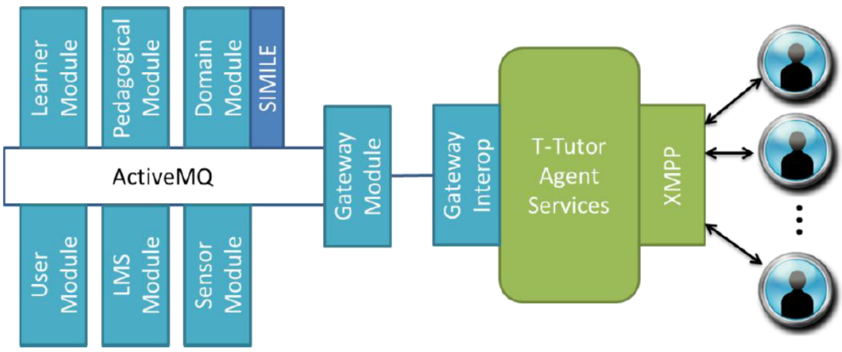

Fig. 9. GIFT and TRADEM Combined Architecture

In the above diagram, the agent services for the TRADEM-Tutor are shown as a plugin to the Gateway interoperability section. These interact with Extensible Messaging and Presence Protocol (XMPP) software, for the purpose of interacting with Google Hangouts or other delivery engine. The use of such architecture allows for the combination of traditional GIFT course elements with the newly added TTutor elements. An example of such an integration may be the use of the Student Information Modules for Intelligent Learning Environments (SIMILE) rule assessment engine for digital games [10], as a practice environment for medical training taught by TTutor.

References

1. Robson, R., Ray, F., Cai, Z.: Transforming Content into Dialogue-Based Intelligent Tutors. The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), vol. 2013. NTSA, Orlando, FL (2013)
2. Ray, F., Brawner, K., Robson, R.: Automating Addie. Intelligent Tutoring Systems 2014 Conference, Honolulu, Hawaii (2014)
3. Ray, F., Brawner, K., Robson, R.: Using Data Mining to Automate Addie. Educational Data Mining 2014, London, UK (2014)
4. Brown, D., Martin, E., Ray, F., Robson, R.: Using Gift as an Adaptation Engine for a Dialogue-Based Tutor. In: Sottilare, R.A. (ed.) Generalized Intelligent Framework for Tutoring Symposium. www.gifttutoring.org, Pittsburgh, PA (2014)
5. Sottilare, R.A., Brawner, K.W., Goldberg, B.S., Holden, H.A.: The Generalized Intelligent Framework for Tutoring (Gift). (2012)
6. Goldberg, B., Brawner, K., Sottilare, R., Tarr, R., Billings, D.R., Malone, N.: Use of Evidence-Based Strategies to Enhance the Extensibility of Adaptive Tutoring Technologies. In: The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC). NTSA, (2012)
7. Gagne, R.M.: Conditions of Learning and Theory of Instruction. (1985)
8. Merrill, M.D.: Component Display Theory. Instructional-design theories and models: An overview of their current status 1, 282-333 (1983)
9. Olney, A., Brawner, K., Pavlik, P., Keodinger, K.R.: Emerging Trends in Automated Authoring In: Brawner, K. (ed.) Design Recommendations for Intelligent Tutoring Systems: Authoring Tools (Volume 3), vol. 3. U.S. Army Research Laboratory, www.gifttutoring.org. In Press. (2015)
10. Mall, H., Goldberg, B.: Simile: An Authoring and Reasoning System for Gift. In: Sottilare, R.A. (ed.) GIFTSym2, vol. 2, pp. 33-42. Army Research Laboratory, Pittsburgh, PA (2014)