Proceedings of the Fifth Annual GIFT Users Symposium

May 2017
Orlando, Florida

Edited by:
Robert Sottilare

Part of the Adaptive Tutoring Series

Proceedings of the 5th Annual GIFT Users Symposium (GIFTSym5)

Proceedings of the 5th Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym5)

Edited by:
Robert Sottilare

Printed in the United States of America
First Printing, May 2017

U.S. Army Research Laboratory
Human Research & Engineering Directorate
Orlando, Florida

International Standard Book Number: 978-0-9977257-1-1

Dedicated to current and future scientists and developers of adaptive learning technologies

CONTENTS

From the Editor

GIFT is a free, modular, open-source tutoring architecture that is being developed to capture best tutoring practices and support rapid authoring, reuse and interoperability of Intelligent Tutoring Systems (ITSs). The authoring tools have been designed to lower costs and entry skills needed to author ITSs and our research continues to seek and discover ways to enhance the adaptiveness of ITSs to support self-regulated learning (SRL).

This year marks the fifth year of GIFT Symposia and we accepted 22 papers for publication. None of this could happen without the efforts of a fantastic team. Our program committee this year did an outstanding job organizing and reviewing, and we want to recognize them for their efforts.

- **Kat Amaya**
- **Michael Boyce**
- **Keith Brawner**
- **Ben Goldberg**
- **Greg Goodwin**
- **Michael Hoffman**
- **Joan Johnston**
- **Jong Kim**
- **Rodney Long**
- **Jason Moss**
- **Scott Ososky**
- **Paul Shorter**
- **Anne Sinatra**
- **Bob Sottilare**

We are proud of what we have been able to accomplish with the help of our user community. This is the fifth year we have been able to capture the research and development efforts related to the Generalized Intelligent Framework for Tutoring (GIFT) community which at the writing of these proceedings has well over 1000 users in over 65 countries.

These proceedings are intended to document the evolutions of GIFT as a tool for the authoring of intelligent tutoring systems (ITSs) and the evaluation of adaptive instructional tools and methods. Papers in this volume were selected with the following goals in mind:

- The candidate papers describe tools and methods that raise the level of knowledge and/or capability in the ITS research and development community
- The candidate papers describe research, features, or practical applications of GIFT
- The candidate papers expand ITSs into previously untapped domains
- The candidate papers build/expand models of automated instruction for individuals and/or teams

The editors wish to thank each of the authors for their efforts in the development of the ideas detailed in their papers. As a community we continue to move forward in solving some significant challenges in the ITS world.

GIFT and the GIFT Symposium will take on a broader perspective as the new Center for Adaptive Instructional Sciences (CAIS) begins formal operations under ARL's Open Campus Initiative. The purpose of CAIS is to encourage the community development of adaptive instructional capabilities & standards. You can learn more about CAIS at https://www.arl.army.mil/opencampus/centers/cais.

Also new this year is GIFT Summer Camp which will pilot in June 2017. GIFT Summer Camp will teach an initial group of GIFT stakeholders how to author adaptive tutors using GIFT. Summer Camp follows on the heels of a successful assessment of the GIFT authoring tools earlier this year. Our intent is to open Summer Camp up to public users in 2018.

Finally, GIFT instructional videos will be available on YouTube this summer.

We would also like to encourage readers to follow GIFT news and publications at www.GIFTtutoring.org. In addition to our annual GIFTSym proceedings, GIFTtutoring.org also includes volumes of the Design Recommendations of Intelligent Tutoring Systems, technical reports, journal articles, and conference papers. GIFTtutoring.org also includes a users' forum to allow our community to provide feedback on GIFT and influence its future development.

Many thanks to all GIFT users…

Bob

Robert A. Sottilare, Ph.D.
GIFTSym5 Chair and Proceedings Editor

Theme I: GIFT Architecture

The GIFT 2017 Architecture Report

Keith Brawner[1], Zach Hylmun[2], and Michael Hoffman[3]
[1]U.S. Army Research Laboratory, [2]Synaptic Sparks, Inc., [3]Dignitas Technologies, LLC.

INTRODUCTION

The first version of the Generalized Intelligent Framework for Tutoring (GIFT) was released to the public in May of 2012. One year later, the first symposium of the GIFT user community was held at the Artificial Intelligence and Education conference in Memphis, Tennessee. Since then, the GIFT development team has continued to gather feedback from the community regarding recommendations on how the GIFT project can continue to meet the needs of the user community and beyond. This current paper continues the conversation with the GIFT user community in a few important ways. The current paper invites and encourages members of the GIFT user community to continue to share their feedback, research findings, and technology innovations with the development team and with one another in order to strengthen the power, usability, and flexibility of the GIFT project. As a follow up to the "GIFT 2015 Report Card and State of the Project" (Brawner & Ososky, 2015), and GIFT 2016 Community Report, the feature requests and responses have been broken out among a number of papers discussing research vectors. This paper discusses the ongoing architectural workings and changes in support of the various sets of projects.

The research and technology innovation efforts presented in the current document include those that are informed by the GIFT user community, and only represent a *fraction* of the overall research, development, and implementation work associated with GIFT. We invite the reader to review the other chapters in this volume, publications on GIFTTutoring.org, and other references described below, to get a sense of the total body of work on the GIFT project. Major themes in this current, 2017 GIFT community discussion include integration with wide-scale systems such as EdX and LearnSphere, further work in enhancing authoring, hosting your own experimental server on Amazon Web Services, and the first GIFT Summer Camp.

Students in UrbanSim conduct operations as a surrogate for the Battalion Commander (BN). The BN analyzes the area of operation (AO) with respect to the stated mission (defined by the Brigade [BDE] Commander), decides on the allocation of resources (by proxy through the player), and assesses progress toward achieving the mission goals. The analysis of an AO is expressed and displayed as a set of interrelated PMESII variables. The role of PMESII values is to organize and aggregate the information received during COIN operations to understand the consequences of previous operations, and assist in planning of subsequent operations. Interpreting PMESII values is a key competence of commanders, and COIN operations are decided and justified in relation to these values, and other intelligence information that may be become available.

GIFT CLOUD

It has been an open secret that the GIFT Cloud instance hosted on cloud.gifttutoring.org has not been a true cloud deployment. A real cloud deployment should have flexible hardware specifications, flexible bandwidth allocations, regular backups, redundancy, and other items. The GIFT Cloud hosted until February had been stationed on a single server computer in Orlando, Florida, subject to power outages and downtime issues. While these issues haven't affected the vast majority of users, the plan was always to move to an Amazon Web Services system. This was performed in February 2017 after some delay. The move from a desktop-based training system to a server-based training system to a cloud-based training system necessitated changes to a number of modules. These changes are described below for posterity.

User and Content Management Systems (UMS/CMS)

Among the primary features missing from a server-based version of GIFT was the ability to upload and manage content to a user account. Previously GIFT supported the idea of a 'classroom training', where a single user has access to all of the content, with no authentication required in order to run the system. The move to a server-based architecture necessitated the management of users, and the separation of content logic from filesystem logic. Moving to this architecture forced the development of both a content abstraction layer and a user abstraction layer. The abstraction layers were created so that a different CMS or UMS could be used at a later time by a developer with a specific requirement. As such, in the event that an organization would prefer the use of a different CMS/UMS, the port of GIFT to use them should be relatively straightforward through the replacement of function calls in the abstraction layer.

As part of the UMS/CMS selection, the following systems were considered: RUSSEL, Exo Platform, Fedora Commons, TYPO3, Moodle, Gooru Learning, EdX, Alfresco, Plone, Jahia, Hippo CMS, Nuxeo, LifeRay, JackRabbit, and Modeshape. Paper space limits a thorough discussion of each of these items, but a list of features and selections is available upon request. Each of these systems was examined for their ability to be a content repository, support different categories of users, differing user roles, sharing of content, built in upload/download, and backend API support. All of the above options are open source, with many used elsewhere in military educational applications. Nuxeo was eventually selected based on its well-developed backend API, Java framework, variety of deployment options, and flexible plugin architecture. Of special interest was the ability to develop plugins to support the individual needs for each Module.

Database

GIFT was initially developed with a mySQL database in place. One of the most notable problems from the first GIFT User's Meeting was the difficulty in the installation of GIFT. Specifically, the use of a mySQL database either: a) required administrator privileges, b) conflicted with a database installation of a previous program with unknown password, or c) was beyond the technical ability of the desired user. Because of these difficulties, the standard database was changed from mySQL to derby in 2013. The derby database program can be embedded within Java applications, needs no installation, and has a small footprint (~2MB), but is not fundamentally equipped to handle server-levels of transactions. The move back to the server necessitated a move to a more fully mature database. This change is invisible to the end user, as GIFT uses postgres in its cloud deployment mode for content, but continues to use derby for the downloadable distribution (both Desktop and Server configuration) and for LMS/UMS functionality as it is intended to be used for classroom-scale experimentation.

Applications

One of the key problems with developing a server-hosted application is that the link to a desktop application is broken. In order to get around this problem, the GIFT program creates a small executable which serves to interface with desktop application. The desktop application, while it does have to have an existing external interface, does not typically need any special configuration. The acceptance of the JWS application is sufficient to allow the server to control the computer-installed application and provide meaningful feedback on events. This includes embedding RapidMiner plugins and other affective modeling software, if appropriate (Rowe, Mott, & Lester, 2015). A similar approach could be taken for Sensor Module components, but has not yet been implemented.

Authentication

Previously, a GIFT version installed on a local machine has no need of authentication to provide services. However, a server-based version must provide a way to distinguish between users. The cloud version of GIFT uses the authentication provided by gifttutoring.org in order to link accounts, content, and access settings. Gifttutoring.org uses the RedMine project as its backend, which uses OAuth for authentication. This authentication can be switched to another (e.g. OAuth) authentication, or with other authentication servers with relatively minimal changes to the source code.

Analysis

One of the key features of GIFT was the ability to run the Event Reporting Tool (ERT). The ERT was a tool developed to look at GIFT log files and determine what actions the student or system had taken during the course of tutoring interactions. This ability was key to fuse multiple streams of sensor data with learning interactions and system interactions in order to answer research questions such as “did the system take an action that lead to learning?” and “what was the learner state when the system made this decision?”. The ERT output results in a format friendly to SPSS for easy analysis, which is also compatible with RapidMiner and most major independent toolsets. Previously, there was not a way to easily port this functionality to a server deployment. The tool has since been reinvented as the “My Research” tab, which allows for both the sharing of tutoring links and analysis of students who use the links.

DOMAIN-INDEPENDENT INSTRUCTIONAL MODELS

GIFT has been constructed in a manner to be independent to the model of the learner, domain, and instruction. As an example of this independent structure, the Domain Module has a configuration called the Domain Knowledge File (DKF), which references code implementations for assessment. Most of these code implementations inherit from an abstract base class of assessment; some of them redirect to external assessment engines, as is the case for the vMedic assessments.

At the time of writing, the Pedagogical Module, at its output, can send four types of request: *Request for Scenario Adaption*, *Request for Performance Assessment*, *Request for Instructional Intervention*, and *Request for Branch Adaption*. Each of these requests, excepting the last, is independent of the instructional model. As a concrete example, consider the following instructional models for adapting within-scenario instruction: a “drill and kill” model, an “expertise based” instructional model, and an “intensive practice” model. The “drill and kill” model runs the same scenario, with feedback, repeatedly until a passing threshold is met. The “expertise based” instructional model runs a scenario, but only provides feedback to novice users; it saves feedback until the end for expert users. The “intensive practice” model restarts after a single mistake, as is common in medical scenarios with high consequence.

Although a simple example, each of the above-discussed models is supported with the current message structure. The *Request for Branch Adaption* message, however, requires the specification of a Merrill Quadrant Enumeration, which is decidedly *not* independent to the instructional model. This model is being expanded to encompass a variety of instructional strategies (passive, active, constructive) within the model, but GIFT is unnecessarily tied to the Component Display Theory during authoring and runtime. This new functionality is scheduled for release.

In order to break the tie to CDT, the Branch Adaption Strategy is going to support an extra layer of abstraction that will allow current and future pedagogical models to recommend different attributes necessary for selecting content. Additionally, during authoring time, it will be possible to select variations among instructional strategies, within boxes. In concrete terms, GIFT currently supports the “Adaptive Courseflow” Course Object for dynamic instruction based on the Engine for Management of Adaptive Pedagogy, which is based on Component Display Theory. GIFT will later support a variety of instructional policy boxes with variations on configurations. Each of these will still be tied to the concepts of instruction, and have access to the same type of information as the Adaptive Courseflow Course Object currently has. More details about these changes are discussed within the paper which discuss Markov Decision Logic (Rowe, Pokorny, Goldberg, Mott, & Lester, 2017) and dynamic After Action Review technologies (Carlin, Brawner, Nucci, Kramer, & Oster, 2017).

ONTOLOGICAL MEDIATION AND VIRTUAL HUMAN TOOLKIT

GIFT has been designed previously as a modular system with modules are interchangeable. The approach of standardization is useful, provided that the community has come to moderate consensus of the operational parts of the system. The ITS community frequently discusses tutoring systems as a byproduct of their instruction, learner modeling, domain modeling, interface to the student, and authorability. While the authoring tools of any system are frequently tied to the product of the system (e.g. Word and .docx, PhotoShop and .psd, iMovie and .imovieproj, etc.), the modeling performed within the system can be modified and changed through the addition of plugins or extensions. The components of an Intelligent Tutoring System (ITS) which have been agreed upon by the community have been implemented as functional modules within the system. If the default Learner Module, Pedagogical Module, and Sensor module are selected, GIFT, as it currently stands, can be used to create intelligent tutoring systems without the development of plugins, without any programming experience, and without the configuration of any files.

With standardization and componentization comes limitation. Some research projects within the research portfolio conduct studies to verify that learning is occurring. Other research projects conduct studies to test different module configurations, instructional selection algorithms, feedback delivery, or other functionality which is captured within existing modular structure. Some research projects have the goal of developing *new* capabilities for a system – capabilities not necessarily agreed upon as needed by the community; capabilities not necessarily essential for the system at runtime. As example of such a capability is a data mining process for developing a policy for selection of review material after a training interaction (Carlin et al., 2016). Which module should encapsulate the data mining portion?

Efforts have been made this year to expand GIFT into an agent-, or service-based system. This change allows for the easier addition of new capabilities, especially new capabilities which fall outside of the scope of existing modules. Examples of capabilities for early integration into this structure include the Markov Decisions Process (MDP) learning from the North Carolina group, the data mining from the Aptima group, and scenario generation techniques.

The move to a service-driven, instead of module-driven, framework additionally allows for the incorporation of external utilities, such as the super Generalized Learning Utilities (superGLU) (Brawner, Goodwin, & Sottilare, 2016), discussed later in this volume. Additional services, such as those provided from the Virtual Human Toolkit (Hartholt et al., 2013) are planned to be added via this approach within the next 12 months. Services include high-quality animation of agents, non-verbal behavior generation, natural language processing, and other items. This approach addressed concerns raised during the previous GIFT Symposium involving RESTful Webservices (Goodwin et al., 2016). These efforts are not planned to be released in the next GIFT release, but will be tested for functionality in the summer of

2017. More about these developments can be requested, or researched through the matching paper in this proceedings (Nye, Auerbach, Mehta, Hartholt, & Fast, 2017).

LEARNING TECHNOLOGY INTEROPERABILITY (LTI) PLUGIN

The Learning Tools Interoperability (LTI) standard was created by the IMS Global Consortium (IMS Global Learning Consortium, 2010). The LTI standard operates on the idea of consumers, such as Moodle and EdX, and providers, such as game-based interactions and homework problems. In this model, a GIFT "course" can operate as a portion of an overall learning interaction. The GIFT course can be developed with standard GIFT tools and linked to the LTI consumer through the sharing of a few simple pieces of information: the CourseID, launchURL, Key, and Shared Secret, as shown in Figure 1. These are then imported as a part of an EdX (or other LTI) course. Data from the shared course can be analyzed with the GIFT "My Research" for the portions of the course that GIFT manages. This is new architectural functionality where none existed prior. More about this functionality can be read later in this work {Aleven, 2017 #1964}.

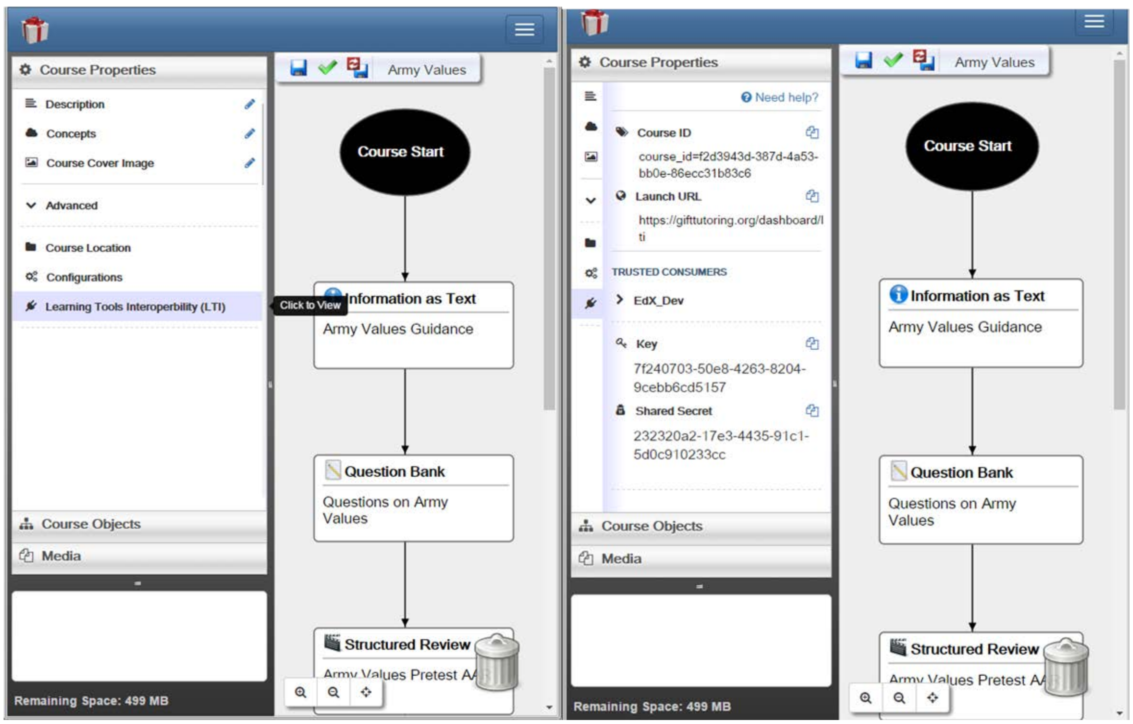


Figure 1 – LTI Interoperability Configuration UI

INCREASED REPORTING

One of the frequently-requested functions in GIFT is the embedding of experience API (xAPI) data (Brawner & Ososky, 2015; Goodwin et al., 2016; Poeppelman, Hruska, Long, & Amburn, 2014). Previously GIFT reported out xAPI data as it matched with simulation-enabled assessments. However, for the purposes of data-mining, storage, and cross-domain synchronization, increased frequency of xAPI reporting has been requested and developed. This increased xAPI reporting has been required for several reasons.

The first of these reasons is for tracking a single user across multiple sessions, such as marksmanship within a classroom, within a simulator, at a live training range, and at a qualification range. Being able to track performance across these items allows for the system to isolate the tutoring actions which not only produced learning in the tutoring system, but eventually led to the end goal of qualification.

The second of these reasons is to enable shared toolsets to make models across multiple domains of instruction, similar to the LearnSphere model. This model allows for multiple systems to store the data in a standard format, such as within a Learner Record Store. Further, reporting xAPI data frequently allows for other, external, systems to use the data, make models of learners or instruction, and share the models with the general GIFT system. For both of these reasons, GIFT now reports xAPI updates as people enter/exit Course Objects, rather than simply at the end of the lesson. Previously, GIFT reported information to the xAPI interfaces less frequently, and at larger gain size.

CONCLUSION AND FUTURE RESEARCH DIRECTIONS

The current deployment of GIFT Cloud is two-fold and split onto a developmental server (dev-cloud.gifttutoring.org) and a production server (cloud.gifttutoring.org). This allows for the rapid prototyping of features in a testable developmental environment before pushing changes cyclically to the production environment. The advantage of this is rapid deployment, while a disadvantage is that GIFT Cloud is now frequently out of sync with GIFT Desktop. Synchronized versions of GIFT Desktop with limited testing/reliability measures are available upon request, or published privately into the SVN.

Future architectural and ontological research is anticipated in a few key areas. The first of these is team tutoring, where a team tutoring version of GIFT has been used in a few private experiments. A team tutor involves a team DKF, team model (instead of learner model), and team pedagogy. Additionally, each individual may have one or more 'role' DKFs, on which they are assessed, in the event that they play multiple roles. The research understandings for each of these items is limited, the architecture functional, and authoring tools non-existent. The second area is in the further breakout of module processes into webservice function calls, as discussed above. The third area of architectural research is how the integration of new training environments can populate authoring tools to enable within-game creation of tutoring content in areas such as the GIFT Wrap project.

Finally, GIFT is intended to provide members of the training, educational, and research communities with the tools and technology needed to efficiently create, manage, and deliver adaptive tutoring content, through leveraging a flexible and extendable framework. GIFT will be continuously improved and developed for the foreseeable future. The authors would like to remind the members of the GIFT community that they have a valuable opportunity to help shape how these and other features are designed and implemented into GIFT. The GIFT development team encourages members of the GIFT community to continue to communicate feedback, issues, suggestions, and results (of research) in order to help us provide the useful tools, powerful technologies, and positive user experiences that will make adaptive tutoring technology accessible to the broadest possible audience.

REFERENCES

Brawner, K., Goodwin, G., & Sottilare, R. (2016). *Agent-Based Practices for an Intelligent Tutoring System Architecture.* Paper presented at the International Conference on Augmented Cognition.

Brawner, K., & Ososky, S. (2015). *The GIFT 2015 Report Card and the State of the Project.* Paper presented at the Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym3), Orlando, FL.

Carlin, A., Brawner, K., Nucci, C., Kramer, D., & Oster, E. (2017). *Educational Data Mining using GIFT Cloud.* Paper presented at the Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym5), Orlando, FL.

Carlin, A., Kramer, D., Nucci, C., Oster, E., Freeman, J., & Brawner, K. (2016). *An Adaptive AAR Capability for GIFT.* Paper presented at the Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym4).

Goodwin, G., Medford, A., Diaz, G., Murphy, J., Twitchell, M., & Blink, M. J. (2016). *Integrating an Interoperable Competency Model with GIFT using the Experience API (xAPI).* Paper presented at the Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym4).

Hartholt, A., Traum, D., Marsella, S. C., Shapiro, A., Stratou, G., Leuski, A., . . . Gratch, J. (2013). *All together now: Introducing the Virtual Human Toolkit.* Paper presented at the International Workshop on Intelligent Virtual Agents.

IMS Global Learning Consortium. (2010). IMS GLC learning tools interoperability basic LTI implementation guide.

Nye, B., Auerbach, D., Mehta, T., Hartholt, A., & Fast, E. (2017). *Building a Backbone for Multi-Agent Tutoring in GIFT.* Paper presented at the Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym5), Orlando, FL.

Poeppelman, T., Hruska, M., Long, R., & Amburn, C. (2014). *Interoperable Performance Assessment for Individuals and Teams Using Experience API*. Paper presented at the GIFTSym2, Pittsburgh, PA.

Rowe, J., Mott, B., & Lester, J. (2015). *Opportunities and Challenges in Generalizable Sensor-Based Affect Recognition for Learning*. Paper presented at the Artificial Intelligence in Education, Madrid, Spain.

Rowe, J., Pokorny, B., Goldberg, B., Mott, B., & Lester, J. (2017). *Toward Simulated Students for Reinforcement Learning-Driven Tutorial Planning in GIFT.* Paper presented at the Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym5), Orlando, FL.

ABOUT THE AUTHORS

Keith Brawner, PhD *is a researcher for the U. S. Army Research Laboratory's Human Research & Engineering Directorate (ARL-HRED), and is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT). He has 11 years of experience within U.S. Army and Navy acquisition, development, and research agencies. He holds a Masters and PhD degree in Computer Engineering with a focus on Intelligent Systems and Machine Learning from the University of Central Florida. His current research is in ITS architectures and cognitive architectures. He* *manages research in adaptive training, semi/fully automated user tools for adaptive training content, and architectural programs towards next-generation training.*

Zach Heylmun *graduated from the University of Florida with a degree in Digital Arts and Science Engineering. After graduation, he worked for Lockheed Martin on low-level, high performance graphics as well as virtual reality rendering for flight simulation and training. Since starting his own company, Voidstar Solutions, as well as helping to form Synaptic Sparks, a 501c3 charity dedicated to STEM education, he has worked with a wider variety of technologies. Through a combination of efforts, both for, and non-profit, he has worked on web technologies, mobile applications, and server infrastructure. He is currently the technical lead for Synaptic Sparks engineering efforts for the GIFT project.*

Michael Hoffman *is a senior software engineer at Dignitas Technologies, and Dignitas's lead GIFT engineer. He has been responsible for ensuring that the development of GIFT meets the evolving customer requirements in addition to supporting both intelligent tutoring for computer based training and intelligent tutoring technology research of the growing user community. Michael manages and contributes support for the GIFT community through various mediums including the GIFT portal (www.GIFTTutoring.org), annual GIFT Symposium conferences, and technical exchanges with ARL.*

Integrating MOOCs and Intelligent Tutoring Systems: edX, GIFT, and CTAT

Vincent Aleven[1], Ryan Baker[2], Nathan Blomberg[3], Juan Miguel Andres[2], Jonathan Sewall[1], Yuan Wang[4], Octav Popescu[1]
[1]Carnegie Mellon University, [2]University of Pennsylvania, [3]Dignitas Technologies, LLC,
[4]Teachers College, Columbia University

INTRODUCTION

Massive Open Online Courses (MOOCs) are successful and widespread, but existing MOOCs have limited capacity to adapt to learners' individual characteristics. They often lack learner models that dynamically capture learner differences and can be used to drive adaptivity. Many have noted that MOOCs have both relatively low completion rates and relatively poor evidence for positive learning outcomes. It is likely that MOOCs would be more effective if they could personalize instruction based on learner characteristics, for example, learners' knowledge or personal interests. They might be more effective, further, if they would provide more learning-by-doing activities (Koedinger et al., 2015).

One path towards making MOOCs more adaptive, with a broader range of learning-by-doing activities, is to integrate existing adaptive learning technologies into MOOCs. In our ongoing project, we integrate the Generalized Intelligent Framework for Tutoring (GIFT) and the Cognitive Tutor Authoring Tools (CTAT) into the widely used edX MOOC platform. GIFT provides a framework and authoring tools for adaptive instruction (Goldberg & Hoffman, 2015; Goldberg, Hoffman, & Tarr, 2015; Sottilare, 2012). The CTAT tool suite can be used to build (among other things) example-tracing tutors, a type of intelligent tutoring system (ITS) that has proven to be robust, classroom ready, and effective (e.g., Aleven et al., 2016b). Like other ITSs, example-tracing tutors support learning-by-doing (practice on complex, recurrent problem types) with adaptive feedback and adaptive problem selection. ITSs have been shown to be very effective in enhancing student learning in a wide range of domains (Kulik & Fletcher, 2015; Ma, Adesope, Nesbit, & Liu, 2014; Steenbergen-Hu & Cooper, 2013; 2014; VanLehn, 2011) and can be authored, increasingly, with efficient and easy-to-learn authoring tools, such as CTAT.

The integration of GIFT and CTAT facilitates the addition of learning-by-doing activities to edX and enhances its adaptivity to learner differences. GIFT and CTAT have different roles to play in this integration, but both would enhance the adaptivity in MOOCs. The types of adaptivity offered by GIFT and CTAT are compatible but complementary, a prime reason why combining them has the potential to benefit learners. GIFT's engine for managing adaptive pedagogy (eMAP) module lets an author create policies for adaptively navigating Merrill's four quadrants (Goldberg et al., 2015). These quadrants characterize four types of instructional activities in a 2x2 grid defined by two dimensions: rules/instances and tell/elicit. EMAP policies for adaptively navigating these quadrants in an individualized manner can consider cognitive factors, metacognitive factors, and other factors. They are primarily outer-loop policies, meaning they take care of task selection (VanLehn, 2006). On the other hand, CTAT tutors offer complex problems with inner-loop adaptive support (VanLehn, 2006), meaning various forms of within-problem guidance, such as feedback on steps and next-step hints. CTAT, together with the TutorShop (Aleven et al., 2016b), a learning management system created in tandem with CTAT and geared specifically to ITS use, also offers outer-loop adaptive support, namely, Cognitive Mastery based on Bayesian Knowledge Tracing (Corbett, McLaughlin, & Scarpinatto, 2000), in which problem selection is individualized based on a student's knowledge. Our project will synergistically combine these different adaptive methods. We will demonstrate this integration and its potential for enhancing student outcomes in the context of the edX MOOC "Big Data in Education: Core Methods in Educational Data Mining" (BDEMOOC).

The principal software development challenges are, first, to integrate GIFT into edX and second, to integrate CTAT/TutorShop into GIFT. To meet these challenges, we take advantage of existing e-learning interoperability standards, namely, the Learning Tools Interoperability (LTI) and the Experience API (xAPI) standards (*IMS Global Learning Tools Interoperability™ Implementation Guide,* 2012; *xAPI Architecture Overview*, 2015). The project builds on our recent work, which has integrated intelligent tutors authored in CTAT into edX MOOCs (Aleven et al., 2015b; 2016a) using the LTI standard. We are currently working to create a first version of the adaptive BDEMOOC by integrating GIFT and CTAT/TutorShop separately into edX, with the course materials integrated into these platforms. This paper reports our work in progress. Specifically, we report on making GIFT an LTI Provider and on extending the content in the BDEMOOC, using the CTAT and GIFT authoring tools. This work addresses the first of the two challenges just mentioned; the second remains for future work.

Integrating GIFT into edX and other LTI Tool Consumers

To integrate GIFT into edX, as mentioned, we rely on the Learning Tools Interoperability (LTI) specification (*IMS Global Learning Tools Interoperability™ Implementation Guide, 2012*). The LTI specification distinguishes application programming interfaces for learning management systems (LMSs) and for learning activity objects or "tools." Here, the LMS is called the LTI Tool Consumer, while systems that provide the learning activities (via URLs) are LTI Tool Providers. In addition to edX, several other popular MOOC and e-learning platforms implement the LTI Tool Consumer interface, including Coursera, Canvas, Moodle and Blackboard.

A first step in our current project, therefore, is to make GIFT's web delivery system an LTI Tool Provider. This first step achieves an initial configuration in which GIFT and CTAT are integrated separately into edX as LTI Tool Providers; this integration already enables enhanced adaptivity in edX, although, as discussed further below, it does not yet achieve the goal of integrating CTAT within GIFT. In a later phase of the project, we will also implement the LTI Tool Consumer protocol in GIFT, for integration with CTAT/TutorShop.

Integration details

We modified the GIFT framework to be an LTI Tool Provider using the LTI 1.1 specification. This GIFT extension makes it possible to embed a GIFT *course* into any LMS or e-learning platform that implements the LTI Consumer interface. Figure 1 shows the components in GIFT that were modified to allow a GIFT course to be run from a Tool Consumer, through an LTI launch request. Step numbers (N) in the following description refer to the circled numbers labeling data flows in the diagram.

To invoke an LTI Tool, from a page referring to LTI content, the Tool Consumer (i.e., the LMS, in this case, edX) (1) triggers an initial LTI launch request configured by the instructor in the Tool Consumer authoring software. The request data includes the reference to the GIFT course that is to be launched.

A new GIFT component (the LTI Tutor Servlet) was created to handle all incoming LTI launch requests. Each request is validated per the LTI specification and (2) the LTI user requesting access is initialized in the GIFT UMS database. Once the LTI user is initialized, the GIFT LTI Tutor Servlet (3) responds back to the Tool Consumer with an appropriate redirect URL. The redirect URL (4) directs the Tool Consumer to the (5) GIFT LTI Landing Page for the user. Once this page is displayed, GIFT (6) receives a request to load the GIFT course for the LTI user. The GIFT Dashboard Web Servlet is used to handle the load course operation. During this request to load the GIFT course, the LTI launch request details (7) are validated again for security purposes. The GIFT File Services API (8) is then used to load the GIFT course for the LTI user. A progress bar shows the user the progress of the course loading process.

After the course is loaded, the LTI user (9) requests to launch the GIFT course. The launch GIFT course request (10) is received by the GIFT Tutor User Interface (TUI) Web Servlet. The LTI launch request details are validated (11) another time for security purposes. Once the LTI launch request details are validated, GIFT (12) presents the LTI user with the started GIFT course in the web client. The LTI user then (13) is able to take the GIFT course until (14) it ends. Once the GIFT course ends, the LTI user is presented (15) with a final LTI Course Ended Page.

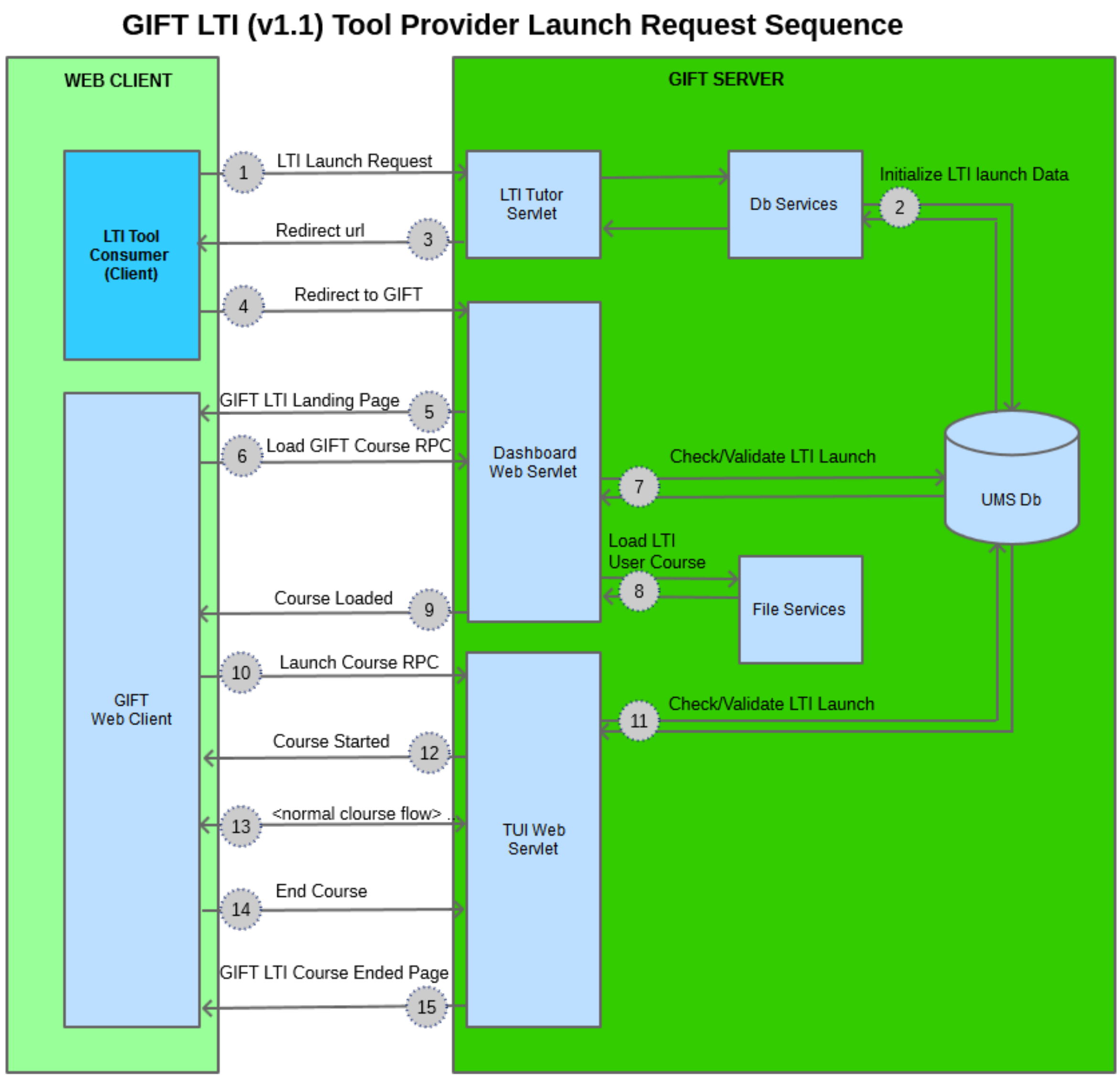


Figure 1. GIFT LTI Tool Provider launch request sequence

Challenges and next steps

We encountered a number of challenges during this effort. One challenge was anonymous users, since LTI users are not authenticated to GIFT as normal GIFT users are authenticated. To solve this challenge, the LTI Tutor Servlet now validates the LTI user and stores the LTI user information in a new database table.

A second challenge was a security issue in the way that GIFT courses were launched. Originally GIFT courses were referenced by a URL containing a human-readable path to the GIFT course file name. When this URL became the address for an LTI launch request, we found that it was exposed to students launching GIFT via LTI. Hence a malicious user could relatively easily access a different course by guessing and entering its file path instead. As a remedy, we replaced the file path in the URL with a universally unique identifier (UUID), which by itself provides no indication of the target GIFT file name or course.

Future integration efforts will provide the capability for GIFT to report a score to the LTI Tool Consumer, via a URL supplied as a parameter when the GIFT activity is invoked. In addition, a GIFT instructor will have the option to collect GIFT data on the users that have taken the GIFT course via LTI.

Adapting GIFT courses to run under edX

The unit of integration for GIFT into an LTI Consumer is an entire GIFT course. This makes it possible to embed any amount of GIFT content in a panel on a single page of an edX MOOC, even just a single exercise. The considerations behind this decision were at least partly technical: each entry to GIFT from edX requires authentication, and in GIFT a student authenticates himself or herself at the whole-course level, not for individual course modules. The result is that an existing GIFT course might need to be largely atomized into individual modules to be appropriate in the context of a MOOC. In our case, we may have whole GIFT courses consisting of just a single Adaptive Courseflow (i.e., EMAP) instance.

Future Work: GIFT as LTI Tool Consumer

In the final version of the MOOC we will create, we will use GIFT to select and deliver which activities and content the students receive, depending on their interest and knowledge, switching between CTAT-authored activities and GIFT-authored activities in the MOOC, and providing remedial conceptual instruction where appropriate. In this way, we combine the adaptive capabilities of GIFT (primarily, but not exclusively, for the outer-loop – the loop over tasks) and CTAT (primarily, but not exclusively, for the inner loop – the loop over steps within a task) in a more tightly integrated fashion than in the initial integration described above.

To integrate CTAT into GIFT, we will again we follow the LTI standards, but this time GIFT will become an LTI Tool Consumer. (As mentioned, CTAT/Tutorshop is already an LTI Provider.) That is, we leverage the LTI standards to enable any GIFT course to include *any* LTI-compliant learning tools, making available to GIFT not only CTAT tutors but also a host of learning activities beyond CTAT. This feature greatly expands the range of third-party-developed learning experiences available to GIFT courses, cuts costs, and eases development by re-using existing products. A sketch of the proposed final integration is in Figure 2. At the simplest level of interaction, CTAT (as it can already, under LTI v1.1) will return scores to GIFT as the student progresses through each exercise; GIFT could use these scores to decide whether or not to present remedial content. More detailed communication between CTAT and GIFT could be mediated by xAPI statements stored in and retrieved from a Learning Record Store (LRS).

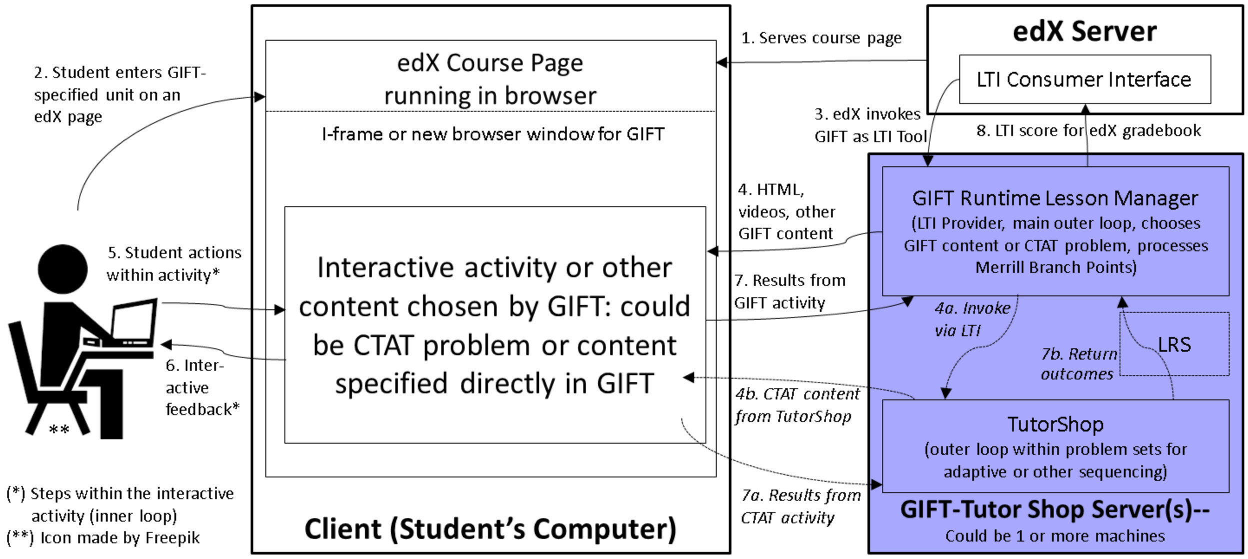


Figure 2: Final integration of GIFT and CTAT-TutorShop in edX, to be achieved in future work.

For example, take a student in the BDEMOOC (the edX course that will become our first adaptive MOOC) who has reported considerable interest in clustering, factor analysis, and bottom-up methods in general—all key topics in this course. This student's efforts will be directed towards these topics. In (say) week 2, rather than continuing further study of prediction models, as would be the default path through the course, the student will begin the study of clustering. Imagine that this student then successfully masters some of the key facts and concepts around clustering but afterwards skips straight to the practical "using the method" assignment in RapidMiner without viewing the videos on procedures for clustering. The student then struggles with the first CTAT assignment on clustering, specifically with respect to selecting how many clusters to use. With data from CTAT, GIFT recognizes that the student needs remedial support and that he or she has covered the facts and concepts. GIFT then uses this information to recommend that the student watch the video on procedures for selecting the number of clusters.

Adapting MOOC content for use with GIFT

The Big Data in Education course (BDEMOOC) has been offered in MOOC contexts across two iterations in the past. Its first iteration was on Coursera, and its second iteration was on edX. This section describes the ideas and the efforts involved in taking the existing curriculum and course materials and modifying them in order to utilize GIFT's and CTAT's adaptive capabilities.

The CTAT tutors embedded in the BDEMOOC ask students to come up with results in RapidMiner from the course's datasets and then query the students' understanding of these numbers. The hints in the tutors' questions, sometimes extending more than 10 levels deep, coach students to help generate their results; the tutors also provide feedback messages to explain and correct common misconceptions. An example of a CTAT tutor running in the BDEMOOC is shown in Figure 3.

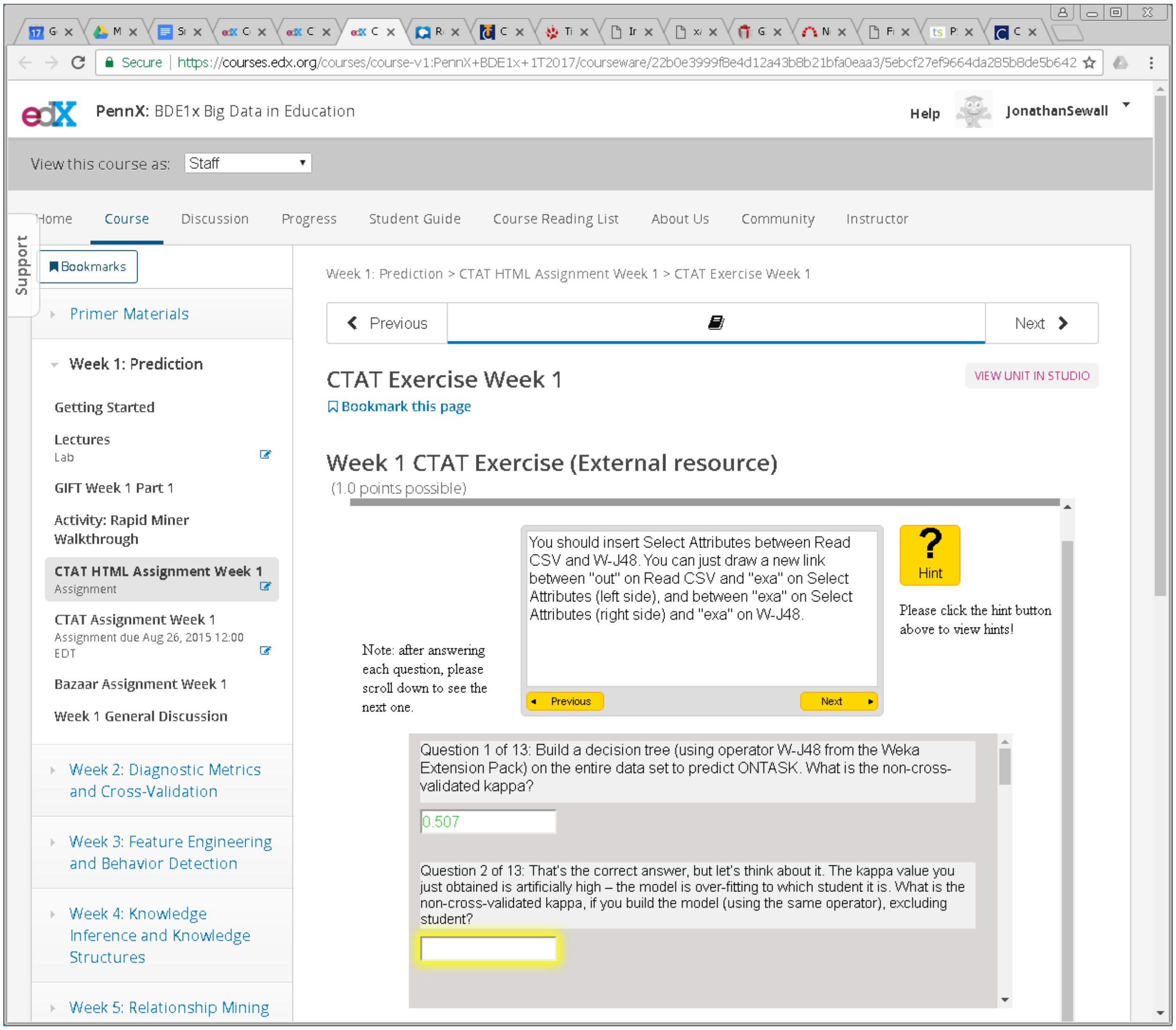


Figure 3: CTAT tutor running in the BDEMOOC. The student has asked for a hint on Question 2 and then requested additional help via the Next button on the Hint window. Because later questions could give the student clues to this answer, they are hidden until the student has answered this question correctly.

Identifying concepts

A major component of GIFT's learner model focuses on the student's competencies and how these are broken down into individual concepts. As such, the first step towards permitting GIFT to adaptively choose content within our course was to identify the individual concepts within the weekly modules. The previous iterations of BDEMOOC were comprised of 8 weekly modules, each having 4-8 video lectures. We identified approximately 200 concepts across all lectures. To give an idea of the granularity of these concepts, those from just the first half of week 1 are shown in Figure 4.

BDEMOOC2017 -- CDT matrix

fx Topic

	A	B	C	D	E	F	G	H
1	Topic	Rules/Exposition	Examples	Recall	Practice	Conceptual	External Resources	
2	Glossary of things we'll cover in more detail later, but you need for {the walkthrough, ba1, ca1}							
3	Course Introduction	V1-1	V1-1	R1-1				
4	Three V's	V1-1 (full re-record)						
5	Diagram of topics in course					Wk 1		
6	What is Prediction	V1-2	V1-2	R1-2		CC1-4		
7	What is Regression	V1-2	V1-2	R1-3, R1-4		CC1-4		
8	Computing a Value for Regression	V1-2	V1-2		V1-2-quiz1			
9	Understanding Contribution of Variable to Model	V1-2			V1-2-quiz2			
10	Variable Transformations	V1-2	V1-2					
11	Benefits of Linear Regression	V1-2	n/a					
12	Risks of Interpretation with Multicolinearity	V1-2	V1-2					
13	Regression Trees	V1-2	V1-2	R1-7				
14	What is Classification	V1-3	V1-5	R1-5, R1-6	W1 (add sao ped	CC1-4		
15	Domain Specificity of Education	V1-3, V1-4	n/a					
16	Variables to Exclude in Classification	V1-3 (slides 49-5	V1-3		A1 (bde asgn.1 should exclude UNIQUEID; UNIQUEID bug; eliminat			
17	Step Regression	V1-3	V1-3		V1-3-quiz-1 (green), A1			
18	Logistic Regression	V1-3	V1-3, V1-5	R1-7, R1-8				
19	Logistic/Step Regression Limitations	V1-3	n/a			CC1-2		
20	Decision Trees	V1-3	V1-3	R1-8	W1, A1			
21	Decision Tree Benefits	V1-3	n/a			CC1-2		

Figure 4: Concepts from lectures in week 1 of Big Data in Education, mapped to Component Display Theory (plus conceptual activities that do not fit clearly in the theory). Specific videos are denoted V*week-video*, Recall items are denoted R*week-item*, Quizzes are denoted V*week-video*-quiz-*number*, Assignments are denoted A*week*, Walkthroughs are denoted W*week*, and Conceptual activities are denoted CC*week-prompt*.

We plan to create new content for BDEMOOC, as well as design and implement adaptive policies, primarily in GIFT, so the BDEMOOC activities can be selected and sequenced according to individual students' needs. In this initial iteration, we plan to take the first step towards making the course more adaptive through the modifications outlined below.

Course reorganization. Since none of the existing BDEMOOC course activities on edX currently have adaptive outer loop control, we will re-organize the course content in terms of Merrill's framework (on which GIFT's EMAP module is based) and embed it into GIFT and edX. These efforts will consist of three types of content creation: First, we will convert existing activities, so they can be used adaptively within GIFT or CTAT. Specifically, we will convert comprehension quizzes (currently existing within edX) to CTAT, divide existing content into four Merrill Quadrants, and remix/redivide videos for use in remediation and adaptive curriculum sequencing. Second, we will create new material for existing topic areas, to fully cover all four Merrill Quadrants (where reasonable; some topics may not lend themselves to practice, for example). This effort includes the creation of recall content. Third, we will create new material for new topic areas, to support deeper personalization of BDEMOOC to individual learners.

Adaption of content and creation of new content. In addition to reorganizing the course, we will bring more adaptivity into the activities within the course, taking advantage of the CTAT tools. Generally, CTAT-built tutors support multi-step problem-solving activities, with step-level support in the form of hint messages – offered on almost every step, to guide students through the thinking processes necessary to produce the correct answer, and buggy messages – adaptive guidance given when students provide a wrong answer that indicates a known misconception. Students working with a CTAT tutor can, at any point in the problem-solving process, voluntarily choose whether to access the hint messages and how many hint messages they would like to view. The initial versions of the CTAT tutors created for the BDEMOOC have been designed in a way that subsequent problem steps or questions will not appear until the student successfully completes the current step. To bring in more adaptivity for these activities, we plan to use the following methods: First, we will break existing tutor problems into smaller chunks and intersperse these smaller chunks between lecture videos. This way, learners will be presented with problem-solving steps relevant to the video lecture immediately preceding it. (By contrast, the current learning sequence has students going through all lecture videos during a week before attempting a single problem assignment at the end of the module.) Second, we will modify the hints in the CTAT-authored tutors to direct the learner to exact points in specific video lectures. In doing so, learners are encouraged to go back to watch the lecture video in order to revisit the learning content that is helpful toward answering the question correctly. The current hint framework allows the students to keep requesting hints until the final hint returns the answer to the question. Finally, in order to support deeper content-level personalization of BDEMOOC, we will create interactive activities for an additional fourteen topics1. These will supplement the current set of ten interactive activities built in CTAT for nine topics.

Running the adaptive BDEMOOC as real-world testbed

The new forms of adaptivity enabled by the GIFT/CTAT/edX integration will be piloted in BDEMOOC. This will provide a demonstration of the technical feasibility of the new forms of adaptivity created in the project. Specifically, in the upcoming years we plan to continue running BDEMOOC in edX at least three times. After each course run, we will iteratively improve and refine the course activities based on analysis of log data (as in the OLI Statistics course redesign based on KC analysis, Lovett, Meyer, & Thille, 2008). For data representing performance within CTAT, sent directly to DataShop, we will use the DataShop's learning curves and Performance Profiler to choose steps with the worst performance and revise those. Data on video watching will be analyzed in relation to assignment performance, in which students work with CTAT tutors, to see if students are watching the videos they need and appropriately using video resources to help them complete assignments. Forum data will be analyzed to determine how positive is students' sentiment to the modifications and what topics students continue to struggle with. In addition, we will gather questionnaire data to learn about how students experienced the adaptive MOOC. Attitudinal surveys will be analyzed to determine student attitudes towards the course.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

By integrating GIFT/CTAT/edX, we are creating a MOOC that synergistically combines three forms of adaptivity: GIFT EMAP adaptive outer-loop control, CTAT adaptive outer loop control, and CTAT adaptive inner loop functionality. We will take advantage of GIFT's adaptive method for navigating the four Merrill quadrants, selecting activities based on individual student characteristics. It is our goal that almost all educational activities will be embedded into GIFT and under GIFT EMAP control, with the exception of the course discussion forum. In the future, when GIFT invokes a CTAT problem set, it will be able to set various parameters that control how CTAT tutors operate within the given problem set. Within these problem sets, CTAT/Tutorshop will support adaptive problem selection, based on Bayesian Knowledge Tracing and Cognitive Mastery – this method has been shown to improve student learning, compared to giving all students the exact same set of problems (Corbett et al., 2000). CTAT will also provide adaptive support for multi-step problem-solving activities, with adaptive hints and feedback that can be sensitive to the student's path through the problem.

So far, we have achieved an initial integration in which GIFT and CTAT activities are embedded separately into edX. To achieve this integration, we made GIFT an LTI Provider. This initial integration

[1] Correlation Mining and Post-Hoc Testing, Causal Data Mining, Association Rule Mining, Sequential Pattern Mining, Differential Sequence Mining, Network Analysis, Epistemic Network Analysis, Clustering, Factor Analysis, Q-Matrices, Partial Order Knowledge Spaces, Bayesian Networks, Bayesian Knowledge Tracing, Performance Factors Analysis.

partially achieves the forms of adaptivity listed above. What is missing is that we do not yet support EMAP control over CTAT activities, which will further enhance the adaptivity (i.e., make it possible to schedule CTAT activities adaptively while also considering GIFT-based activities; this gives us two different authoring tools, each with their own strengths and limitations, to author activities in the different quadrants). In order to achieve GIFT EMAP control over CTAT tutor assignments, we will further extend GIFT so it implements the LTI Consumer protocol.

Our project will provide a practical framework for adaptivity in MOOCs, supported by proven authoring tools. The project will lay out a path for creating future GIFT/CTAT/EdX courses. Beyond creating such a MOOC, we will develop guidelines and a case study paper on how to effectively do this process going forward. Interestingly, the same LTI-based integration can be reused for embedding CTAT tutors or other LTI-compliant learning objects into GIFT courses (without edX), as well as for adding GIFT-based adaptivity (without CTAT) to edX courses or other LTI-compliant MOOC platforms, e-learning platforms, and LMSs.

The work will make a theoretical contribution by investigating how multiple instructional models can be combined: adaptive traversal of Merrill's quadrants, tutored problem solving by an ITS, and standard MOOC pedagogy focused on individual, self-regulated learning with a variety of online resources (video lectures, multiple-choice questions with automated feedback to test your understanding, practice problems with peer feedback, and online discussion with peers in forums). Because tutors and MOOCs were developed separately, their synergy is underexplored. Questions that we will investigate through this project include: What leverage is there in enabling CTAT to communicate its student model to GIFT, and how could GIFT use it to make better adaptive sequencing decisions? We will also study how the learner experience is impacted by this newer, more flexible pedagogy for MOOCs. We will survey learners to compare their attitudes towards the revised course to the attitudes reported in previous iterations of the course (both in terms of reported attitudes, and survey completion), and we will conduct sentiment analysis on discussion forum data to see how attitudes towards the course are impacted.

REFERENCES

Aleven, V., Sewall, J., Popescu, O., Ringenberg, M., van Velsen, M., & Demi, S. (2016). Embedding intelligent tutoring systems in MOOCs and e-learning platforms. In A. Micarelli, J. Stamper, & K. Panourgia (Eds.), *Proceedings of the 13th international conference on intelligent tutoring systems, ITS 2016* (pp. 409-415). Springer International Publishing.

Aleven, V., McLaren, B. M., Sewall, J., van Velsen, M., Popescu, O., Demi, S., . . . Koedinger, K. R. (2016b). Example-Tracing tutors: Intelligent tutor development for non-programmers. *International Journal of Artificial Intelligence in Education*, *26*(1), 224-269.

Aleven, V., Sewall, J., Popescu, O., Xhakaj, F., Chand, D., Baker, R. S., . . . Gasevic, D. (2015). The beginning of a beautiful friendship? Intelligent tutoring systems and MOOCs. In C. Conati, N. Heffernan, A. Mitrovic, & M. F. Verdejo (Eds.), *Proceedings of the 17th international conference on artificial intelligence in education, AIED 2015* (pp. 525-528). New York: Springer.

Corbett, A., McLaughlin, M., & Scarpinatto, K. C. (2000). Modeling student knowledge: Cognitive tutors in high school and college. *User Modeling and User-Adapted Interaction*, *10*, 81-108.

Goldberg, B., Hoffman, M. (2015). Adaptive Course Flow and Sequencing through the Engine for Management of Adaptive Pedagogy (EMAP). In *Workshop on Developing a Generalized Intelligent Framework for Tutoring (GIFT): Informing Design through a Community of Practice* (pp. 46-53), held in conjunction with the 17th International Conference on Artificial Intelligence in Education (AIED 2015).

Goldberg, B., Hoffman, M., & Tarr, R. (2015). 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 adaptive intelligent tutoring systems* (Vol. III, Authoring Tools, Chap. 26, pp. 319-334). Orlando, FL: US Army Research Laboratory.

IMS Global Learning Tools Interoperability™ Implementation Guide (Final Version 1.1). (2012, March 13). Retrieved from https://www.imsglobal.org/specs/ltiv1p1/implementation-guide

Koedinger, K. R., Kim, J., Jia, J. Z., McLaughlin, E. A., & Bier, N. L. (2015). Learning is not a spectator sport: Doing is better than watching for learning from a MOOC. In G. Kiczales, D. M. Russell, & B. Woolf (Eds.), *Proceedings of the second ACM conference on Learning @ Scale* (pp. 111-120). New York: ACM.

Kulik, J. A., & Fletcher, J. D. (2015). Effectiveness of intelligent tutoring systems. *Review of Educational Research*, 0034654315581420.

Ma, W., Adesope, O. O., Nesbit, J. C., & Liu, Q. (2014). Intelligent tutoring systems and learning outcomes: A meta-analysis. *Journal of Educational Psychology*, *106*(4), 901. doi:10.1037/a0037123

Sottilare, R. (2012). Considerations in the development of an ontology for a generalized intelligent framework for tutoring. In I3M defense and homeland security simulation conference (DHSS 2012).

Steenbergen-Hu, S., & Cooper, H. (2013). A meta-analysis of the effectiveness of intelligent tutoring systems on K–12 students' mathematical learning. *Journal of Educational Psychology*, *105*(4), 970-987.

Steenbergen-Hu, S., & Cooper, H. (2014). A meta-analysis of the effectiveness of intelligent tutoring systems on college students' academic learning. *Journal of Educational Psychology*, *106*(2), 331-347.

VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education*, *16*(3), 227-265.

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, *46*(4), 197-221.

xAPI Architecture Overview. (2015, October 23). Retrieved from https://www.adlnet.gov/adl-research/performance-tracking-analysis/experience-api/xapi-architecture-overview

ABOUT THE AUTHORS

Vincent Aleven *is an Associate Professor in Human-Computer Interaction at Carnegie Mellon. He has 25 years of experience in research and development of ITS based on cognitive theory and self-regulated learning theory, with a focus on K-12 mathematics. He and his colleagues have created the CTAT/Tutorshop ITS authoring environment, which non-programmers can use to create adaptive tutors. These widely-used tools improve the cost-effectiveness of tutor authoring by a factor of 4-8. Aleven has over 200 publications to his name. He and his colleagues and students have won 7 best paper awards at international conferences. He is or has been PI on 9 major research grants and co-PI on 10 others.*

Ryan Baker *is Associate Professor at the University of Pennsylvania, and Director of the Penn Center for Learning Analytics. Baker has developed models of student engagement in over a dozen online learning environments, and has led the development of an observational protocol and app for field observation of student engagement by over 150 researchers in 4 countries. His Massive Online Open Courses have served approximately 96,000 learners. He was the founding president of the International Educational Data Mining Society, and is currently serving as Associate Editor of two journals. Baker has co-authored published papers with over 250 colleagues.*

Nathan Blomberg *is a senior software engineer at Dignitas Technologies with over fifteen years of professional software development experience. Nathan holds a B.S. in management computer systems from the University of Wisconsin Whitewater. After graduation, Nathan spent over 12 years working with commercial software development in the video game industry helping create high profile AAA franchises including EverQuest, NCAA college football, and the Madden football games. He contributed to notable* *companies such as Sony Online Entertainment and Electronic Arts. Nathan joined Dignitas Technologies in 2014, where he is currently a developer for the GIFT and Marksmanship projects.*

Juan Miguel Andres *is a PhD student at the University of Pennsylvania, under the Teaching, Learning, and Teacher Education program. He is a member of the Penn Center for Learning Analytics, where he currently researches the generalizability of findings on student engagement and course completion in MOOCs.*

Yuan Wang *is a Doctoral Research Fellow in Cognitive Sciences at Columbia University Teachers College. Her current research focuses on Massive Online Open Courses (MOOCs) learner motivation and prediction of learner post-course development, with the goal of building frameworks toward operationalizing MOOC learner success. Specifically, her projects take a comprehensive approach by linking three sources of learner data: pre-course learner motivation, within-course learner engagement, and post-course development. Particularly, measurement of post-course development reflects both individual learner career development as well as advancement of communities of practice. Her recent publications are in Journal of Online Learning and Teaching and Journal of Learning Analytics.*

Octav Popescu *is a Senior Research Programmer/Analyst at CMU's Human-Computer Interaction Institute, working on the Cognitive Tutor Authoring Tools project.*

Jonathan Sewall *is a project director at CMU's Human-Computer Interaction Institute, working on the Cognitive Tutor Authoring Tools.*

Building a Backbone for Multi-Agent Tutoring in GIFT (Work in Progress)

Benjamin D. Nye[1], Daniel Auerbach[1], Tirth R. Mehta[1], Arno Hartholt[1]
University of Southern California, Institute for Creative Technology[1]

INTRODUCTION

As intelligent tutoring systems (ITS) increasingly need to interoperate and co-exist, emerging systems have transitioned toward service-oriented designs to enable modularity and composability of tutoring components made and/or maintained by different research and development groups. However, as a research community, we have still not reached a point where it is trivial for a new service to be added into a system like the Generalized Intelligent Framework for Tutoring (GIFT; Sottilare, Goldberg, Brawner, & Holden, 2012). In an early paper considering this issue with respect to the GIFT architecture (Nye & Morrison, 2013), we proposed addressing this issue by building toward a lightweight multi-agent architecture where certain services act as autonomous agents: "*a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to affect what it senses in the future*" (Franklin & Graesser, 1997; p. 25).

In our work in progress described here, we discuss how we are approaching the opportunity to build such capabilities into GIFT. The high level goals of our work are targeting two core goals for GIFT: A) to be a lightweight framework that will expand access to and use of ITS and B) to help GIFT to increase the intelligence and effectiveness of its services based on data over time. We are currently targeting the first goal, which will underpin the second goal. However, what does it mean to be a lightweight framework? In this context, a "lightweight framework" is framed as minimizing the following criteria: (1) hardware requirements, (2) software expertise to design services, (3) software expertise to use existing services, (4) software expertise to stand up the message-passing layer between agents, and (5) a minimal working message ontology (Nye & Morrison, 2013). Since our original paper four years ago, GIFT has made significant strides in reducing barriers related to hardware by building a cloud-based version and software expertise to use GIFT services through authoring tools. It has also developed a growing ontology of messages (e.g., https://gifttutoring.org/projects/gift/wiki/Interface_Control_Document_2016-1). With that said, despite now-extensive documentation, designing new services for GIFT is still not trivial and strong expertise is required to pass messages between GIFT modules and agents (either internal or external).

To address these issues, the Building a Backbone project is working toward agent-oriented designs that build on GIFT's existing service-oriented framework. By moving from services toward agents, modules will be able to act more autonomously, enabling capabilities such as plug-and-play, hotswapping, and selecting between multiple services providing the same capabilities. These new capabilities are intended to reduce barriers to building new GIFT-compatible services and also to integrating GIFT with other service-oriented ecosystems. The first steps toward these capabilities are an ontology mapping service and an initial integration that combines GIFT, the Virtual Human Toolkit core framework for agents, and the SuperGLU framework for adding agent-oriented capabilities for coordinating services. This paper reports on work to date, with an emphasis on target capabilities, design decisions, challenges, and open research questions for this work.

TARGET CAPABILITIES FOR GIFT AS A MULTI-AGENT ITS

When considering the advantages for GIFT as a multi-agent framework, we must consider the question: advantages for who? What stakeholders benefit and how? GIFT has at least six different types of users:

- *Learners (Use Services):* Students and trainees who participate and learn from GIFT courses
- *Instructors (Use Services):* Deliver, modify, and possibly design GIFT courses
- *Basic Course Designers (Configure Services):* Modify or build course content with wizards/tutorials, including selecting or configuring a group of services verified to work together.
- *Advanced Course Designers (Compose Services):* Build advanced content and adaptive behavior, by selecting and configuring services to work together as a group
- *Service/Agent Programmers (Make/Add Services):* Code new services or agents used by GIFT
- *Framework Providers (Combine Service Ecosystems):* Maintain and interface other large frameworks with GIFT

The above list indicates a stakeholder type, their expected role for interacting with services/agents, and then a brief description their primary goal and role when dealing with GIFT. In terms of direct effects, the latter three stakeholders who build and compose the services can benefit from greater modularity, re-usability, and interoperability. By lowering barriers to entry, agent-oriented design should encourage a growing ecosystem of intelligent and interactive services. A comparison can be made with modern web design, where seemingly simple HTML pages actually involve interacting JavaScript and frames hosted by dozens of web services that each specialize in particular functionality (e.g., ad hosting, static content delivery, feeds of weather information). From the standpoint of less-technical stakeholders, the benefits are indirect and would be evident by a broader and more effective set of adaptive courses. For Learners and Instructors, a successful multi-agent ITS should be "magic" in that it just works and is effective without requiring any knowledge about what agents, services, and frameworks do under the hood. For Course Designers, the standard is "plug-and-play" where advanced capabilities such as services and agents should have reasonable defaults and clear boundary conditions/suggestions for when they can be used. For Service Programmers, the goal is that new agents and services can be quickly integrated to communicate with core GIFT services. For Framework Providers, the goal is to make it straightforward to complete either a one-time or iterative integration that enables a suite of existing services (with its own messaging/communication) to interact with GIFT *and any other services or frameworks that already work with GIFT* (i.e., GIFT as a hub for interoperability).

To accomplish this, a multi-agent framework needs to satisfy certain criteria:

1. Agent Communication Language(s): Services should communicate using explicit messages, where every service has a clear set of messages that it sends and receives (i.e., a message set analogous to an asynchronous API). Services do not necessarily need to use the same message set, provided that some mapping is available to map messages between ontologies.
2. Plug-and-Play: Services must not be hard-coded to send messages to any specific other service, but instead must either send messages with no explicit recipients (i.e., blind sending) or must resolve their specific service communication at runtime (e.g., proposal patterns, negotiation).
3. Independent Agency: Every service should be designed to contain no assumptions or minimal assumptions about how other agents process information internally, and instead rely on messaging to determine declared or inferred states about other services it interacts with. This lends itself to declarative messaging, where instead of a standard command of "Start <x>" and

assuming that <x> is started, a service might instead message “Request Start <x>” and watch for a message of “Inform <y service> Start <x>.”

4. Network Structure/Protocol Independent: Services should be able to plug in using a single protocol rather than requiring a separate API for every protocol (e.g., Websockets, AMQP, HTML5 postMessage), so that the content and handling of messages is decoupled from how the message is transported. Related to this, it must be assumed that some services may need to pass messages through multiple systems or protocols to reach their final destination, but should not require any knowledge of the network structure or protocols to do their job, unless their job is specifically-related to communication protocols.
5. Graceful Failure: Services should have certain goals (implicit or explicit) that they attempt to achieve, based on the best information available (i.e., messages received). In the case of a violation of expected communication (e.g., requests receive no replies, an ill-formed message is handled), each service should continue to try to achieve its goals. Errors and poorly-behaving services should be recorded to enable fixing or removing problem services.

For a shorter summary of these principles, it could be said that services for a system like GIFT should act like humans do as agents in a well-run team: we know what job and goals we are trying to do, we do not mind-read (i.e., internal processes of other humans) or need to know the whole organizational structure (e.g., the network), and if other people fail to give us what we need to do our job then we still do our best and/or report those people as a problem for the team. For some services, such as reactive and stateless calculations, these principles are easy to follow. For others, such as an agent in a complex simulation, it can be more complex. However, without these principles, agents become strongly coupled to either other services or to fragile factors irrelevant to their actual goals (e.g., network structure). A multi-agent architecture following these principles offers many advantages over traditional designs:

- ***Specialization:*** In a multi-agent system, agents can specialize in tasks. While this benefit also applies to other component-based systems, agents can communicate their abilities using semantic messages and agent communication languages (Bellifemine et al., 2008).
- ***Decentralization:*** Modern software environments rely on a large number of interacting components, where even a single web page is actually an orchestration of many web services, domains, platforms (e.g., desktop vs. mobile), and a mixture of various client and server-side code. ITS need to follow these patterns, through service-oriented and agent-oriented designs.
- ***Customization:*** An implication of distinct agents that communicate using messages is that each agent does not rely on the specific internal state of any other agent. This means that each agent can be rapidly customized, provided that it still uses the expected messages.
- ***Data-Driven Enhancements:*** Related to customization, this means that it is also possible for agents (and simpler services) to use collected data to improve their performance against certain metrics and goals.

With this said, GIFT is not a project starting from the ground up: it is an existing, substantial framework that already follows certain software architecture patterns and has certain assumptions. To add these capabilities while retaining its existing strengths is an ongoing research project, which will be described in the following section.

HIGH LEVEL DESIGN

In this first year of the project, the stakeholder use-cases that we are focusing on are *Framework Providers* and *Service Programmers*. To build toward the case of integrating new framework providers into GIFT, we are building a branch of GIFT that integrates services from an upcoming open source release of Virtual Human Toolkit (Hartholt et al., 2013) by using the SuperGLU (Generalized Learning Utilities) framework (Nye et al., 2014). These three frameworks each have unique strengths and architectural principles. Of these, SuperGLU's role is to provide fundamental framework components to abstract away issues related to plug-and-play capabilities, network structure independence, translating between agent communication languages, and service agency. By comparison, GIFT and the Virtual Human Toolkit are both longer-standing frameworks with substantial existing services and content are being connected as an example of integrating a framework provider (i.e., the Virtual Human Toolkit). The complementary capabilities of these frameworks are that:

GIFT (Generalized Intelligent Framework for Tutoring; Sottilare et al., 2012) is an open-source ITS architecture that implements a service-oriented architecture for tutoring, which can be deployed as a standalone installation, a network client/server architecture, and a cloud-based framework. Through its functionality for authoring tools and ITS course package managers, GIFT offers both fine-grained capabilities (e.g., services for student modeling) and larger-grained tools.

The **Virtual Human Toolkit** is a set of interacting services that primarily focus on training systems where highly-realistic animated agents act as mentors or act out interactive scenarios as virtual roleplayers. The Virtual Human Toolkit contains a number of services, including animated agents (typically rendered in Unity), the nonverbal behavioral generator (NVBG) to automatically determine gestures based on speech, and SmartBody to coordinate agent behavior (e.g., locomotion, lip syncing). In addition to its primary services, Virtual Human Toolkit also provides message patterns that help for registering and controlling services, leading it to be used as a messaging framework to coordinate behavior of other services in ITS, such as goal-oriented agents, natural language processing and audio-visual sensor streams (Kenny et al., 2007). The Virtual Human Toolkit is based on the SAIBA framework (Vilhjálmsson et al., 2008) and utilizes several messaging standards (Kopp et al., 2006; Heylen et al., 2008; Scherer et al., 2012). It underpins a number of ITS and training systems (Campbell et al., 2011; Kenny et al., 2007).

The **SuperGLU** framework is an ongoing open source project designed for real-time coordination of different ITS services, based on multi-agent communication. SuperGLU branched out of the Memphis team for the Office of Naval Research (ONR) STEM Grand Challenge, where it was implemented as the Shareable Knowledge Objects ITS (SKO-ITS) architecture (Nye et al., 2014). This system uses explicit semantic message-passing, where messaging is based on two standards: the ADL xAPI standard for reporting learning experiences (Murray & Silvers, 2013) and the FIPA agent-communication language standard (FIPA, 2013). This framework uses special "gateway" services to abstract away the device and network architecture (e.g., client/server; cross-domain messaging). This enables each individual service to focus entirely on the messages that it sends and receives, rather than being programmed with knowledge about the specific services generate the messages. This system is currently being used for the ONR ElectronixTutor project, where it is being used to integrate four separately-developed ITS from Worchester Polytechnic Institute, the University of Memphis, Raytheon-BBN, and Arizona State University (github.com/GeneralizedLearningUtilities/SuperGLU). A pre-prototype version of the SuperGLU architecture supported the NewtonianTalk project, which integrated AutoTutor for dialog-based tutoring, Physics Playground for simulation-based learning, and GIFT (Ventura et al., 2015). SuperGLU currently has implementations of its core functionality in Python, JavaScript, and Java.

These three frameworks offer distinct advantages that will be leveraged when combined into a multi-agent framework. GIFT, with its high-level capabilities for course management and pedagogical strategies, offers an integrator architecture that can leverage new services that plug into the system. It also provides significant infrastructure for tutoring simulation-based training.offers a bridge to a larger ecosystem of existing services and tools (e.g., high-fidelity tutoring agents and avatars) which will continue to expand as new applications use and build upon an open source release of the Virtual Human Toolkit (OSVHT). Finally, the SuperGLU framework will support rapid creation and integration of new services, with a particular emphasis on web-based ITS, cloud-based services, and multi-agent systems. Integrating the SuperGLU framework and its expanding set of services with GIFT will provide a variety of specialized services and tools that accomplish common ITS tasks, ranging from cloud-based storage (e.g., the GLUDB project; github.com/memphis-iis/GLUDB) to service wrappers for HTML animated agents (Nye et al., 2014). Based on this integration, our work consists of four phases of functionality: 1) Framework Interoperability (combining existing services and ontologies), 2) Service Design (creating and adding a new service), and 3) Service Composition (composing multiple services to work together), and 4) Advanced Agent Capabilities.

Framework Interoperability (Functionality to Combine Frameworks/Services)

The current work on this project is building basic Framework Interoperability capabilities, focusing on the use-case using the GIFT, SuperGLU, and the Virtual Human Toolkit these frameworks together. This phase will culminate in a proof-of-concept reference implementation where a GIFT course interacts with an OSVHT animated pedagogical agent who reacts to behavior on multiple websites that communicate with GIFT in real-time using the SuperGLU framework (depicted in Figure 1). A proof-of-concept GIFT mini-course will demonstrate (visually and programmatically) the functionality and capabilities of this integration, along with any helper services to streamline this process for future GIFT course designers. We plan to have this mini-course to present the topic of phishing, through interacting frames that attempt to trick the user into providing information through insecure methods (e.g., http vs. https) or to the incorrect site. This will be facilitated by using SuperGLU to coordinate communication between iframes on multiple domains.

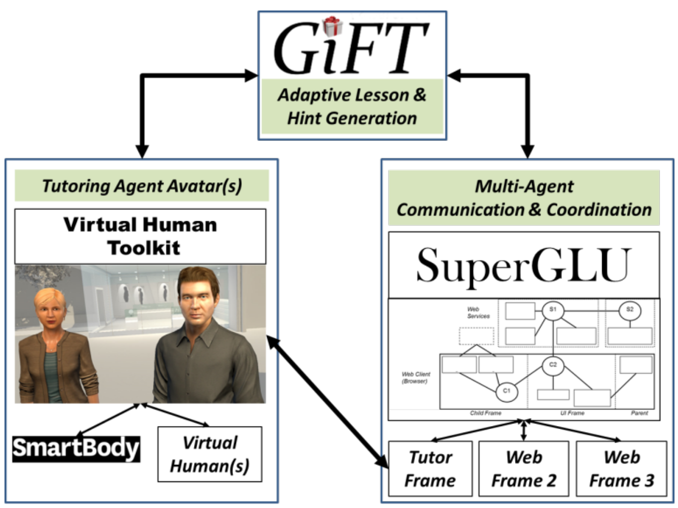


Figure 1: Concept of First Integration Use-Case

Service Design (Making and Adding New Services)

Based on the basic capabilities put in place for the first phrase of research, we will create user-facing tutorials and documentation about how to perform certain common multi-agent ITS behaviors, such as building a Virtual Human tutor, creating a new cloud-based SuperGLU service, or using SuperGLU to integrate cross-domain services. To improve and evaluate the usability of this from the service design standpoint, we will recruit approximately novices (e.g., students) to complete these tutorials and revise instructions based on their feedback and observations on their performance. If successful, a student should be able to build and add a service to GIFT with a level of effort similar to a class homework assignment.

New agents and services will be integrated either using wrappers developed based on the SuperGLU framework that plug in to GIFT's services (e.g., an adaptor pattern) or by services that directly adhere to GIFT's messaging protocols (e.g., ActiveMQ). For client-side communication (e.g., browser-based services), SuperGLU JS libraries allow rapid integration of multiple cross-domain iframes that communicate using HTML5 messaging and can also communicate with server-side services (e.g., via Websockets).

Service Composition (Selecting and Combining Services)

While programming and adding a single service is something that is straightforward to train for a one-off addition to a system like GIFT, this quickly becomes more complex if you imagine dozens of programmers creating agents which add either new or alternative capabilities (e.g., alternate models to support pedagogical model decisions). This is a service composition problem, where tools must exist to determine which services should be used by a course, to modify default configurations, and to validate that services coordinate meaningfully (e.g., that some other service should be able to send a message required to make another service work effectively). Authoring for agents will rely on the GIFT authoring tools to the greatest degree possible. This requires some level of authoring support in GIFT for Advanced Course Designers, and as such requires careful design consideration before adding such capabilities.

Advanced Agent Capabilities (Ontology Mapping, Hotswapping, Negotiation)

After adding the baseline capabilities for Service design, research will also focus on enhancing the multi-agent communication layer that integrates GIFT, Virtual Human Toolkit, and SuperGLU. The goal is to extend light-weight communication patterns for ITS and other advanced learning technologies. While not all agent communication patterns are appropriate for ITS, a number of existing multi-agent patterns would be particularly beneficial for ITS. The capabilities that we wish to foster in the GIFT ecosystem are:

- *Hot-Swapping:* Adding, removing, or switching services at runtime, which are then discovered by other ITS agents and used (or ignored) appropriately.
- *Service Competition:* Real-time competition between different agents that provide the same information but that might be more appropriate or predictive for different learner types, domains, or contexts (e.g., mobile vs. desktop).
- *Ontology Mapping:* Services that facilitate communication between systems that communicate using messages, but which whose messages use different structures or labels.

Agent communication patterns that can support these goals are agent proposals, agent negotiation, and brokering services, as shown in Figure 2. (Bellifemine et al., 2008). Proposal patterns allow a service to send out a request to a network of agents, who then respond with proposals to do the task (and possibly additional meta-data about performance criteria). The original requesting agent may then select a proposal and submit a firm request to the agent. Proposal patterns can resolve the issue where multiple agents or services can provide the same capabilities, so that a requesting agent needs to prioritize or select one service over another. Negotiation patterns are an extension of proposal patterns, which allow for multi-turn communication to revise a proposal until it meets the goals and requirements of both agents.

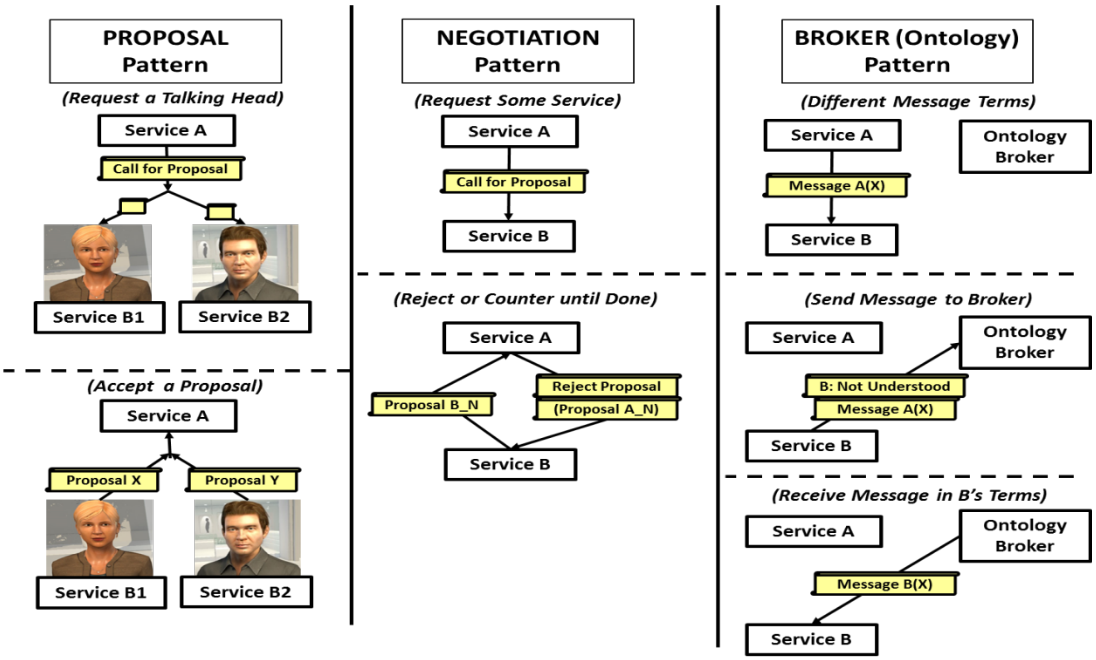


Figure 2: Examples of Key Agent-Communication Patterns for ITS

Brokering agents are a second known-important category of agents, which can be used to store information that is useful to a variety of other agents (FIPA, 2013). The most critical of such agents for Framework Integration is an "ontology broker" to resolve messages would not be understood by some agents or frameworks. An ontology broker stores the semantic messaging definitions and mappings between different ontologies for messages. As such, the ontology broker can attempt to translate a message from a format that the receiving agent cannot understand into one that it can act upon. As part of the first phrase (Framework Interoperability), we have built a prototype ontology converter that focuses on syntactic and semantic mappings between messages from different ontologies (in this case, certain messages shared in common by GIFT,OSVHT, and SuperGLU).

These capabilities would be particularly important for DoD training in future blended learning, augmented reality, and on-the-job training environments. Such environments contain different sensors and different contextual factors (e.g., network availability) that will influence the types of information streams and AI inference that are available. A key benefit of these advanced patterns will be context-based enhancement of training (e.g., taking advantage of capabilities when available) and soft-fail systems (e.g., using lower-fidelity ITS services when more advanced ones are unavailable or ineffective).

CURRENT STATE: FEATURES BEING INTEGRATED INTO GIFT

Work thusfar on the Building a Backbone project has focused on the fundamental problem of Framework Interoperability. In particular, our current progress is intended to simplify two barriers to integrating services and frameworks with GIFT: adding external services and ontology mapping capabilities.

Mapping Between Messaging Ontologies

Mapping between messages from different ontologies is done using a two-stage process. The first step is the identifying that a received message is valid message type to convert into a given target message type (i.e., the expected type for the output of the conversion). In this step, an incoming message checked to determine if it is a valid message type that has been registered in the ontology converter (e.g., a GIFT, SuperGLU, Virtual Human Message, etc.). Once it is ensured that the message is a known type, the system searchers for a Mapping, i.e. a message type that it could successfully mapped onto. This is done using an OntologyMapping object which allows registering mappings, which in this case has registered a limited subset of OSVHT<->SuperGLU and GIFT<->SuperGLU messages that have reasonable equivalents (approximately 15 total for GIFT<->SuperGLU and only one OSVHT<->SuperGLU at present).

In the second step, a series of conversions are applied that transform the original message to the target message type. In some cases, this may be a single conversion (if a direct mapping exists) while in other cases multiple conversions might be required (e.g., in this case, no mappings have been specified for OSVHT<->GIFT, but since certain messages from each ontology have equivalents in SuperGLU, they can be converted through two sequential transforms). The goal of this service is to enable registering semantic message mappings, either at load-time or on-the-fly, that enable information and requests sent from one service or framework to communicate with other services and frameworks. These converters could also be set up as OntologyBroker services which themselves communicate explicitly using messages, to enable agents to resolve messages from remote, centralized converters when needed.

Adding Services from an External Framework

For our first minimal use-case, Virtual Human services were integrated with GIFT by communicating through messages on GIFT's ActiveMQ messaging hub. Under normal conditions, these two frameworks would not communicate because neither sends any messages known to the other framework. To communicate between these systems, a SuperGLU gateway was added to a special Agent Container module in GIFT. This gateway contained an Ontology Converter, as shown in Figure 3. Services in Figure 3 that are built-in to GIFT (i.e., Java services) are shown connected with solid lines, while dashed lines represent connections to services that start up through separate processes. The example case was for the Virtual Human to speak a message from GIFT, using appropriate body-language generated by the OSVHT Non-Verbal Behavioral Generator (NVBG) service and text-to-speech by its TTSRelay service.

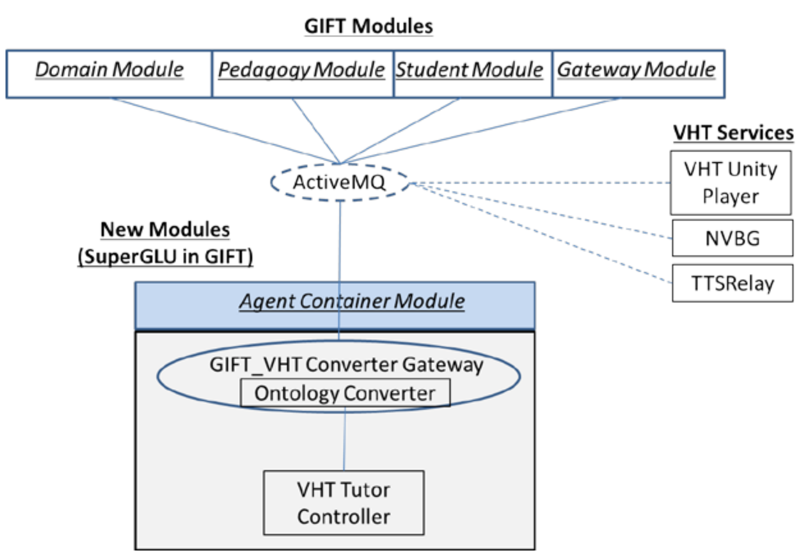


Figure 3. Minimal Integration Example Between GIFT and OSVHT, using SuperGLU

The steps that occur when the message is sent from GIFT to the Virtual Humans toolkit are
1. Agent Container Module listens to messages for all modules, relaying them to *GIFT_VHT Converter Gateway* (implemented in GIFT using a Java port of the SuperGLU library). Most messages that reach the *GIFT_VHT Converter Gateway* converter are ignored because they are have no relevant mapping in the VHT message ontology (which is its target).
2. GIFT sends a *GIFT:Display Guidance Tutor Request* from the Domain Module.
3. The *GIFT:Display Guidance Tutor Request* is received by the *GIFT_VHT Converter Gateway*.
4. The gateway's built-in ontology converter recognizes that a valid mapping chain exists for this message to create an VHT message.
5. The ontology mapping converts the message into a *SuperGLU:Speech* message. The converter then maps the *SuperGLU:Speech* message into a *VHT:vrExpress* message.
6. The *VHT:vrExpress* message reaches the *VHT Tutor Controller*, which modifies the message to set any required optional parameters (e.g., the name of the speaking agent). The OSVHT Tutor Controller currently exists to maintain values for default parameters.
7. The *VHT Tutor Controller* sends a new *VHT:vrExpress* message back to the *GIFT_VHT Converter Gateway*. The gateway does not convert it before passing it to ActiveMQ, because relays OSVHT messages.
8. The NVBG receives the *VHT:vrExpress* message through ActiveMQ. NVBG generates non-verbal behaviors that match the words given, and both the speech and behavioral markup are sent to the OSVHT Unity Player. The tutor says something in the OSVHT Unity Player, by transmitting messages to the TTSRelay to generate speech while the accompanying animations are executed by the character in the OSVHT Unity Player.
While simple, this basic example provides the foundation for more complex service designs.

NEW CAPABILITIES FOR GIFT TUTORING

When considering these additions to GIFT, the question must be asked, "How do these capabilities make GIFT more effective?" While in earlier sections this question was considered at a theoretical and conceptual level, this section considers the advantages and rationale for when GIFT developers should use these capabilities as compared to existing design patterns that GIFT offers. While the focus will be placed on current capabilities being worked on, the next set of capabilities being worked on this year will also be discussed briefly.

The primary rationale the first phrase of this project is to lower barriers to entry for adding new services to GIFT and to demonstrate that functionality by integrating an existing significant framework (the Open Source Virtual Human Toolkit). To understand the benefits of what has been done so far, we must first note the capabilities that already exist in GIFT. Releases of GIFT include three mechanisms for adding external functionality to GIFT: 1) Gateway interop plugins, 2) Domain module conditions that call external assessment engines (e.g., SIMILE; Mall & Goldberg, 2014), and 3) Software branches/forks that modify or extend GIFT as an open source project. Gateway Interop Plugins are the most similar, in that they are designed as interface for external training environments to communicate with GIFT (primarily with the Domain Module). Add-on assessment modules such as SIMILE also have some similarities, in that they represent a submodule plugged in as part of an existing module (in this case, the Domain Module). Finally, at least for GIFT standalone clients (e.g., those not hosted on the GIFT cloud version), new services can be added by simply coding new modules and adding them to a GIFT software build.

In short, GIFT has some established patterns for connecting to other services. Our research here attempts to generalize these patterns. For example, while various training systems can connect as Gateway Interop plugins, these plugins are expected to provide messages that are handled by the Domain Module or by a hard-coded system like SIMILE that provides more fine-grained metrics based on training messages.

There is currently no straightforward way for a training system pass data through GIFT to arbitrary third-party service(s) for further analysis before returning that information to GIFT in an actionable state. In existing GIFT builds, it is unclear where to put novel services that process non-standard messages (e.g., non-GIFT messages, new GIFT-formatted messages) that do not always receive or produce GIFT messages. Second, a system like SIMILE that acts as a subcomponent of a module is currently an explicitly-coded option, rather than part of a larger standardized system of services that allow a module to delegate functionality. Finally, existing GIFT modules assume that they communicate either explicitly through Active MQ messages or sometimes receive information implicitly through other mechanisms (e.g., HTML page choices and submissions). However, as GIFT expands, it will need to pass messages through other protocols to services, such as between other web services and clients (Websockets, REST) or even between HTML iframe services (HTML5 postMessage).

The multi-agent capabilities are being added to GIFT address these challenges, providing new opportunities for developing services for GIFT:

1. AgentContainer Module for Novel Services: First, this work adds a new module to GIFT called the AgentContainer. While this container has no innate functionality, it is special in that it listens to all messages that pass through the ActiveMQ channel so that attached services can process messages and send responses. This container can support services that do not naturally fall into the capabilities of any particular module, including both built-in GIFT services and remote services that are linked in to GIFT as services.

2. Agents in GIFT Core Modules: Second, agents are being made available to add to existing GIFT modules to provide extra capabilities and functionalities. While certain elements of this system are still being designed, the goal would be that these agents would only have access to the same information of the module as a whole, but might override or (in the case of multiple agents) suggest different options to the same situation. This functionality would give a general pattern for implementing certain existing capabilities, such as SIMILE for the Domain Module or work to add reinforcement learning policies to optimize decisions by the Pedagogy Module (Rowe, Frankosky, Mott, Lester, Pokorny, Peng, & Goldberg, 2016).

3. Expanding Gateway Protocols: The SuperGLU library gateways integrated into GIFT are designed to support general-purpose messaging across different protocols. So then, while the existing Gateway Interop Plugin module will still handle connections to many training systems (particularly simulations which are able to contact an ActiveMQ service), the AgentContainer supports building generic gateways to handle other messaging protocols. Currently, a Websocket protocol for real-time bidirectional messaging with web clients (socket.io) is being implemented that will support message passing to and from arbitrary HTML pages opened by GIFT.

4. Communicating with Dynamic or Third-Party Web Pages: On the web-client side, Javascript libraries for Websocket connections to a server (e.g., GIFT) and HTML5 postMessage cross-domain messaging have already been implemented. As such, this work will expand the ability to interoperate with dynamic web pages and cross-domain applications that integrate multiple services on the client side. This means that GIFT courses could be designed to open arbitrary third-party web pages that communicate with a service in the AgentContainer, to enable interoperability with existing frameworks for authoring unique ITS tasks, simulations, or visualizations.

5. Ontology Mapping: As noted above, the ontology mapping capabilities enable gateways added to the AgentContainer to declare how messages in a certain format can be translated into other formats (e.g., GIFT, SuperGLU, VHT, etc.). These are implemented such that messages can be

converted before they are sent to remote systems or after they reach third-party destination. In either case, the same set of declarative mappings can be executed to convert messages.

6. OSVHT Integration: Adding open source VHT interoperability to GIFT may also be useful to future authors. While the current build implements the functionality required to control a pedagogical agent (i.e., non-verbal behavior, text-to-speech control, an animated agent), the VHT framework contains other functionality for natural language processing, dialog controllers, and related functionality. While the current work does not make that entirely trivial (new mappings would need to be created), it establishes the fundamental interoperability that should make future integrations simpler.

7. Expanded Analytics: Finally, while the implications of integrating multiple message ontologies are still being explored, one likely opportunity will be to log messages from a variety of systems in greater detail (e.g., web page event logs, third-party service logs that previously could not be sent to GIFT). Since all ActiveMQ messages in GIFT are logged for later analysis, converting messages from external services into GIFT messages would enable deeper after-the-fact processing, even if those messages only represent generic logging messages. With that said, the benefits of this new information would depend on the goals of the researchers and services logged.

While these capabilities are still being tested, refined, and documented with good examples and tutorials, they represent the first wave of functionality provided by the multi-agent architecture approach. The second wave of functionality will focus on issues related to plug-and-play capabilities (e.g., adding services to a GIFT course elegantly and without dealing with any GIFT source code), hotswapping (e.g., removing and adding agents at runtime based on certain conditions), and patterns to choose between services that provide the same functionality (e.g., proposals, negotiation).

CHALLENGES AND OPEN RESEARCH QUESTIONS

Fundamentally, the challenge of this work is to leverage relatively complex software design patterns (i.e., agent-oriented services) to make the overall experience and ecosystem of GIFT easier and more versatile. In working toward this goal, a substantial challenge is the large differences in skill-sets and use-cases between distinct user groups and use cases. For some users, substantial effort might be warranted provided it only needs to occur once (e.g., integrating a large third-party framework with GIFT and building a full ontology mapping of its messages to GIFT messages). In other cases, any effort beyond normal course activities may be an impossible barrier to cross (e.g., an instructor with limited preparation time trying to add or modify brief course). As such, any multi-agent design must give flexibility for technical users while remaining robust and invisible when interacting with end-users. This is a non-trivial undertaking that raises the question: how can the complexity of software agents be managed such that agents make GIFT *lighter* (i.e., easier to use and maintain) rather than heavier (i.e., harder to understand and configure). As such, a key question is how much of the agent functionality should be exposed to different user types and how such transparency versus opaqueness can be managed.

A second major challenge is that opening the GIFT ecosystem to distributed services introduces challenges for reliability, security, accountability, and debugging. Reliability is impacted because more services and machines introduce greater opportunities for failure, either due to software, hardware, or network errors. Latency can also be a potential issue for services that are distributed across multiple networks and systems: by lowering barriers to add multiple external services, cumulative latency might start to impact responsiveness or performance. Similar problems might be observed with even agents communicating on the same machine with GIFT, if their performance is slow (e.g., a slow agent reducing overall system performance). Security is impacted because information and messages need to cross potentially multiple systems, frameworks, and services within those frameworks. This introduces questions about how trust and credibility should be managed when sending and receiving messages, since even services within the same framework on the same machine might require different levels of authentication and access (e.g., should only be able to exchange certain message types with GIFT). Accountability is likewise more complex in a distributed context: if a service fails for a course, who should be called to fix it? The Course Designer for including it in their course (and might be using it improperly)? The Service Designer who made the service originally (and maybe did not document a limitation that is failing)? GIFT for allowing the service? These are non-trivial problems that require considering analogous situations that support effective ecosystems of services. Finally, debugging is also complex in a distributed, agent-based environment: it is not always clear which agent is failing and the internals of an agent might not always be available to analyze. Worse, the problematic component may not be fixable (e.g., a service from an enterprise framework that requires backwards compatibility). While these challenges may be a ways down the road, and are in some respects good problems to have (e.g., indicate a growing ecosystem) they are also ones where intelligent early decisions can make a positive impact. As such, there are substantial research problems ahead for this work which impact both the technical and social processes of developing and using ITS.

REFERENCES

Bellifemine, F., Caire, G., Poggi, A., Rimassa, & G. (2008). JADE: A software framework for developing multi-agent applications. *Information and Software Technology, 50*(1), 10–21.

Campbell, J., Core, M., Artstein, R. Armstrong, L, Hartholt, A., Wilson, C.A., Georgila, K.,…& Yates, C. (2011), Developing INOTS to Support Interpersonal Skills Practice, In *IEEE Aerospace Conference*.

Foundation for Intelligent Physical Agents (2013). FIPA ACL message structure specification. www.fipa.org/specs/fipa00061/SC00061G.html.

Franklin, S., Graesser, A.: Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In: Intelligent agents III agent theories, architectures, and languages, pp. 21–35. Springer Berlin Heidelberg (1997).

Hartholt, A., Traum, D., Marsella, S. C., Shapiro, A., Stratou, G., Leuski, A., ... & Gratch, J. (2013). All together now. Introducing the Virtual Human Toolkit In *Intelligent Virtual Agents* (pp. 368-381).

Heylen, D.K.J., Kopp, S., Marsella, S.C., Pelachaud, C., Vilhjálmsson, H.H. (2008) The Next Step towards a Function Markup Language. In *Intelligent Virtual Agents.*

Kenny, P., Hartholt, A., Gratch, J., Swartout, W., Traum, D., Marsella, S., & Piepol, D. (2007). Building interactive virtual humans for training environments. In *I/ITSEC*, 7105 (pp. 174-200).

Kopp, S., Krenn, B., Marsella, S., Marshall, A., Pelachaud, C., Pirker, H., Th´orisson, K., Vilhj´almsson, H., (2006) Towards a Common Framework for Multimodal Generation: The Behavior Markup Language *Intelligent Virtual Agents vol. 4133, pp. 205–217*.

Mall, H., & Goldberg, B. (2014). SIMILE: an authoring and reasoning system for GIFT. In *2nd Annual GIFT Users Symposium*. pp. 33-42.

Murray, K., & Silvers, A. (2013). A learning experience. *Journal of Advanced Distributed Learning Technology, 1*(3-4), 1-7.

Nye, B. D., Graesser, A. C., & Hu, X. (2014). AutoTutor in the cloud: A service-oriented paradigm for an interoperable natural-language ITS. *Journal of Advanced Distributed Learning Technology, 2*(6) 35-48.

Nye, B. D. & Morrison, D. M. (2013). Toward a Generalized Framework for Intelligent Teaching and Learning Systems: The Argument for a Lightweight Multiagent Architecture. In *Artificial Intelligence in Education (AIED) 2013 Workshop on GIFT,* (pp. 105-115).

Rowe, J., Frankosky, M., Mott, B., Lester, J., Pokorny, B., Peng, W., & Goldberg, B. (2016). Extending GIFT with a Reinforcement Learning-Based Framework for Generalized Tutorial Planning. In *Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym4)* (p. 87-97).

Scherer, S., Marsella, S.C., Stratou, G., Xu, Y., Morbini, F., Egan, A., Rizzo, A., Morency, L. (2012), Perception Markup Language: Towards a Standardized Representation of Perceived Nonverbal Behaviors. In *The 12th International Conference on Intelligent Virtual Agents*

Sottilare, R. A., Goldberg, B. S., Brawner, K. W., & Holden, H. K. (2012). A Modular Framework to Support the Authoring and Assessment of Adaptive Computer-Based Tutoring Systems (CBTS). In *I/ITSEC, vol. 2012* (1). NTSA.

Ventura, M., Hu, X., Nye, B. D., & Zhao, W. (2015). NewtonianTalk: Integration of Physics Playground and AutoTutor using GIFT. In *AIED 2015 Workshop on Developing a Generalized Intelligent Framework for Tutoring (GIFT)* (p. 23-29).

Vilhjálmsson, H. and Thórisson, K.R, (2008) A Brief History of Function Representation from Gandalf to SAIBA. In *Proceedings of the 1st Function Markup Language Workshop at AAMAS.*

ABOUT THE AUTHORS

Benjamin D. Nye, Ph.D. *is the Director of Learning Sciences at USC ICT. He received his Ph.D. in Systems Engineering from the University of Pennsylvania in 2011 and previously served as a Research Professor at the University of Memphis. Ben's major research interest is to identify best-practices in advanced learning technology, particularly for frontiers such as distributed learning technologies (e.g., cloud-based, device-agnostic) and socially-situated learning (e.g., face-to-face mobile use). Ben's research tries to remove barriers development and adoption of intelligent tutoring systems so that they can reach larger numbers of learners, which has traditionally been a major roadblock for these highly-effective interventions.*

Daniel Auerbach *is a Research Programmer at USC ICT, who has developed and expanded a variety of intelligent tutoring systems and frameworks. He received a B.A. in Linguistics from Cornell University.*

Tirth Rajen Mehta *is a Student Research Worker at USC ICT, who is working on his M.S. in Computer Science at the USC Viterbi Department of Computer Science.*

Arno Hartholt *is the Director of Research and Development Integration at the University of Southern California Institute for Creative Technologies where he leads the virtual human integration and central asset production & pipeline group. He is responsible for much of the technology, art, and processes related to virtual humans and related systems, in particular at the interchange between research and industry capabilities. He has a leading role on a wide variety of research prototypes and applications, ranging from medical education to military training and treatment. Many of the ICT virtual human capabilities are freely available to the academic research community and US Military through the Virtual Human Toolkit (vhtoolkit.ict.usc.edu). Hartholt studied computer science at the University of Twente in the Netherlands where he got his Master's degree.*

THEME II:
GIFT AUTHORING

The 2017 Overview of the GIFT Authoring Experience

Scott Ososky
Oak Ridge Affiliated Universities
U.S. Army Research Laboratory

INTRODUCTION

One of the primary goals of the Generalized Intelligent Framework for Tutoring (GIFT) is to reduce the time and skill required to create adaptive training. Achieving those goals within the GIFT platform is enabled by a set of authoring tools and associated resources, known collectively as the Authoring Experience. The current paper will discuss the state of the GIFT Authoring Experience, as well as how feedback and data from the community is informing user-centered design efforts within the GIFT authoring tools.

The discussion will focus specifically on work conducted over the past year, including improvements to UI and workflows, as well as additional features that have been added to the authoring tools. Those enhancements will be described in the context of field research, conversations with the user community, and case studies of GIFT use in real-world settings. Finally, this paper will preview aspects of the GIFT authoring experience that are under investigation, and the authoring tool enhancements that are intended to result from ongoing user-centered design research efforts. This paper is intended to benefit both new and experienced GIFT users, and may be of interest to anyone conducting design research or developing interfaces and user experiences for computer-based productivity tools.

NEW AUTHORING TOOLS

The new authoring experience that was described in last year's GIFT User's Symposium proceedings is now available (Ososky, 2016a; Ososky & Brawner, 2016) at GIFT Cloud (cloud.tutoring.org). This new experience is centered providing a unified interface, making language within the authoring tools more consistent and intuitive, and building user-centered tools that support authors' goals. Readers interested in the history and evolution of the GIFT authoring tools are encouraged to read the references contained within this paragraph (Ososky, 2016b; Ososky & Sottilare, 2016).

Specifically, the new authoring experience revolves around a visual course building interface within the Course Creator. From within this interface, all other core aspects of course authoring are accessible to the user. The course flow timeline interface was redesigned based on a flow chart (or discrete event process) metaphor with simple drag-and-drop functionality. The visual structure of the course more accurately suggests the sequencing functions that are available to course authors. The design intent was to evoke a *mental model* of similar, more familiar interfaces in order to make this authoring task more intuitive for novice users (Figure 1). Available course objects are displayed in the toolbox on the left-hand side of the interface. Authors can drag and drop objects onto the timeline in any position. Objects already on the timeline can be re-ordered or deleted as needed.

Some course objects have been renamed to provide a better indication of their functionality and/or breadth, respectively. Complementary to the course object interface is the existence of an on-demand help window that appears in the lower-right hand corner of the interface. Currently, interacting with any of the course objects within the interface displays information about that object within the window. This is useful for new authors, as well as more experienced authors who are trying to decide between multiple viable course objects to complete their course sequencing goals.

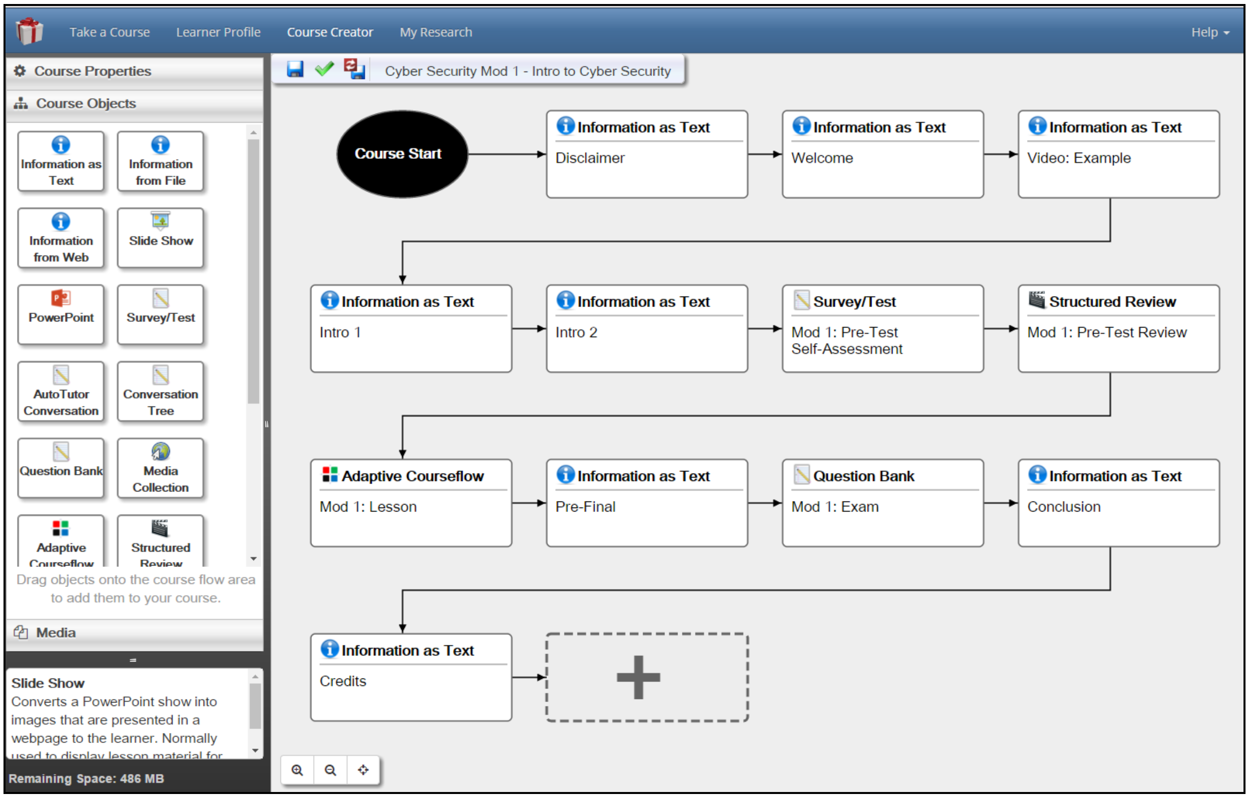


Figure 1: Visual course flow editor uses familiar interaction metaphors to allow authors to quickly create course outlines.

The efficient editing of course objects is also central to the new authoring experience. Recognizing that authoring an adaptive tutor required complex yet usable authoring tools, our design approach to efficiency was inspired by the similarly complex design video game development tools (Lightbown, 2015). The new authoring experience adopts a design philosophy that reduces the number of pop-ups on screen, and keeping multiple pieces of information in view in order to reduce the memory/recall burden on the author. To that end, editing a course object is as simple as clicking it on the timeline. Now, instead of a pop-up, the course object editor opens in a side frame on the right-hand side of the workspace (Figure 2).

Each course object has a different set of editing and configuration options; some editors require more screen space than others in order to be in full view of the author. Therefore, each of the three primary *panels* of the Course Creator are resizable. The course object editing UI can also toggle into *full screen* mode as the authoring work shifts from sequencing to configuring course objects. The editing area, by default, displays the most recent course object that the author has clicked in the timeline. Individual course object editors can be *pinned* to the editing frame and quickly accessed as a series of tabs (like a web browser) should the author need to view or edit multiple course objects simultaneously.

The object that is currently in-focus within the editing frame is also highlighted on the timeline with an animated blue-dashed outline around the object. The intent of that design is to help the author make the connection between what is being edited and where that object exists in the timeline. This is useful when courses have multiple objects that appear similar to one another on the timeline, such as an informational message, or a survey / test (Figure 2).

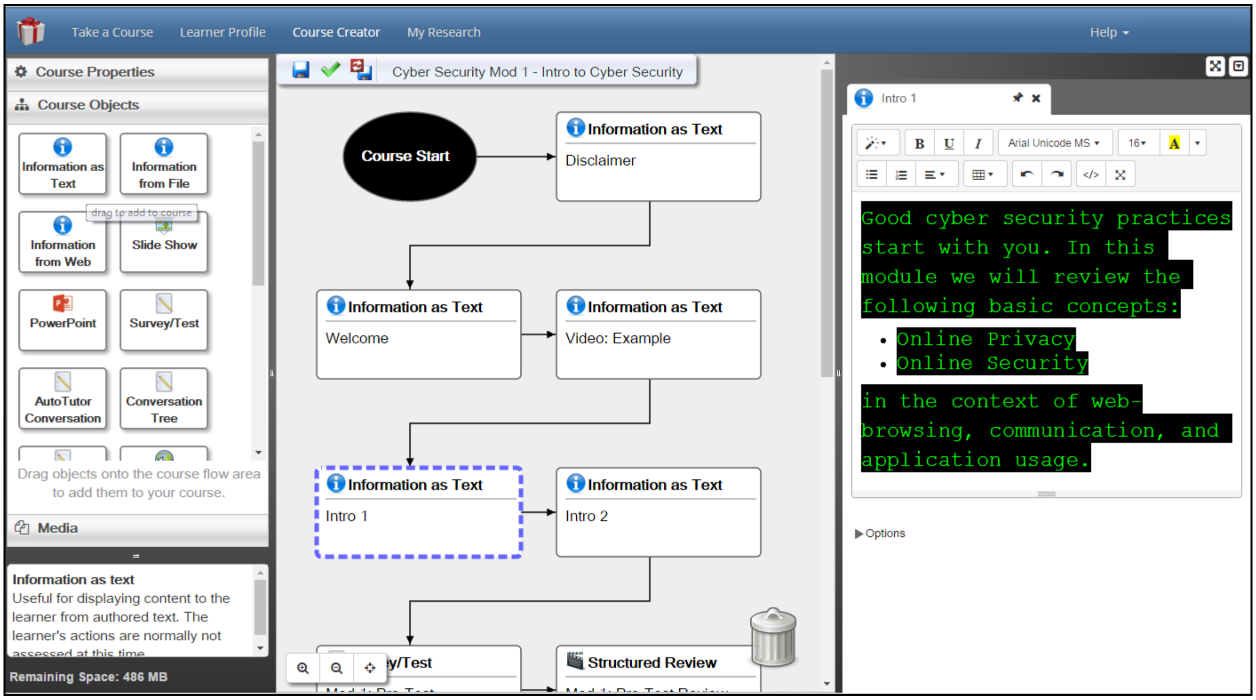


Figure 2: The editing frame appears when interacting with objects on the timeline. All three sections of the Course Creator interface are resizable, based on user needs and preferences.

The survey authoring interface has also been completely re-designed. Specifically, we focused on reducing users' physical effort within the system, or *excise*, by reducing the number of mouse clicks and keyboard commands required to build an individual question. Further, system-level configuration parameters, such as the collection of surveys for a particular course (i.e., survey context) are now automatically managed by the GIFT and are invisible to the author.

The survey authoring interface is now presented in a WYSIWYG style format (Figure 3). After writing the question text, and a first response item, additional response items are automatically added so that the author can quickly complete the question, and then easily get a sense of how that question will look when the learner sees it. Text fields within the survey composer also use the same, familiar rich-text editor that is found within the Information Course Object types. They survey composer now supports quickly copying and moving questions within the interface, further increasing authoring efficiency. Configuration options are organized within different parts of the interface in order to maintain authoring efficiency and the visual integrity of the survey layout. Options such as "force response" or "multi-select" now appear in a side frame, which is dynamically updated as individual questions are selected. The survey system also supports multi-select for batch operations on groups of questions.

Scoring mode (Figure 4, top-middle) is a new feature of the survey composer intended to reduce authors' cognitive workload through progressive disclosure of information. The new mode functions as a toggle between *writing* surveys and *scoring* surveys. Activating scoring mode temporarily locks the questions for editing; the UI adds a series of score boxes to each response for a particular question (Figure 4). Further, if the survey is linked to a set of learning concepts, additional options will appear on each question that will allow the author to associate the question with a concept(s) and set a difficulty for that question. Toggling between the Writing Mode and Scoring Mode reduces visual clutter, and allows authors to focus on one specific aspect of survey composition. The benefit of the two modes is most apparent when questions have complex scoring mechanisms, such as a matrix of responses.

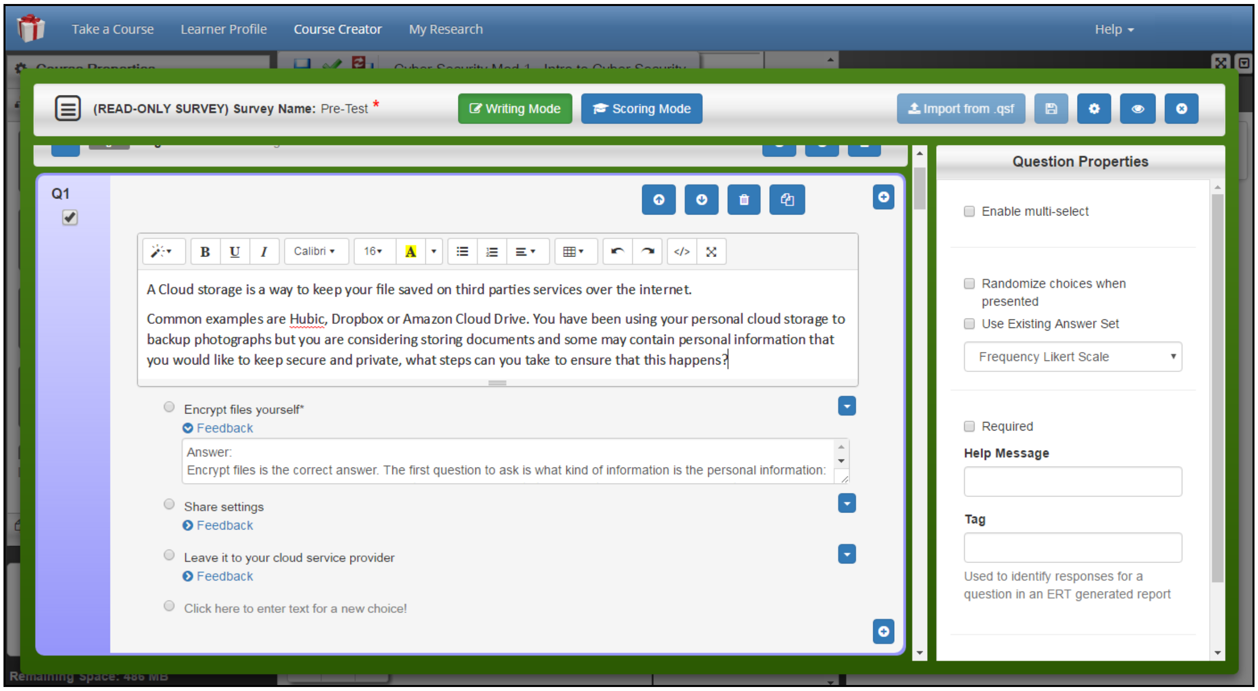


Figure 3: The survey composer has been completely redesigned as a WYSIWYG editor with rapid survey creation in mind. Configuration options appear on the right-side frame. Scoring options are enabled through a separate scoring mode toggle at the top of the interface.

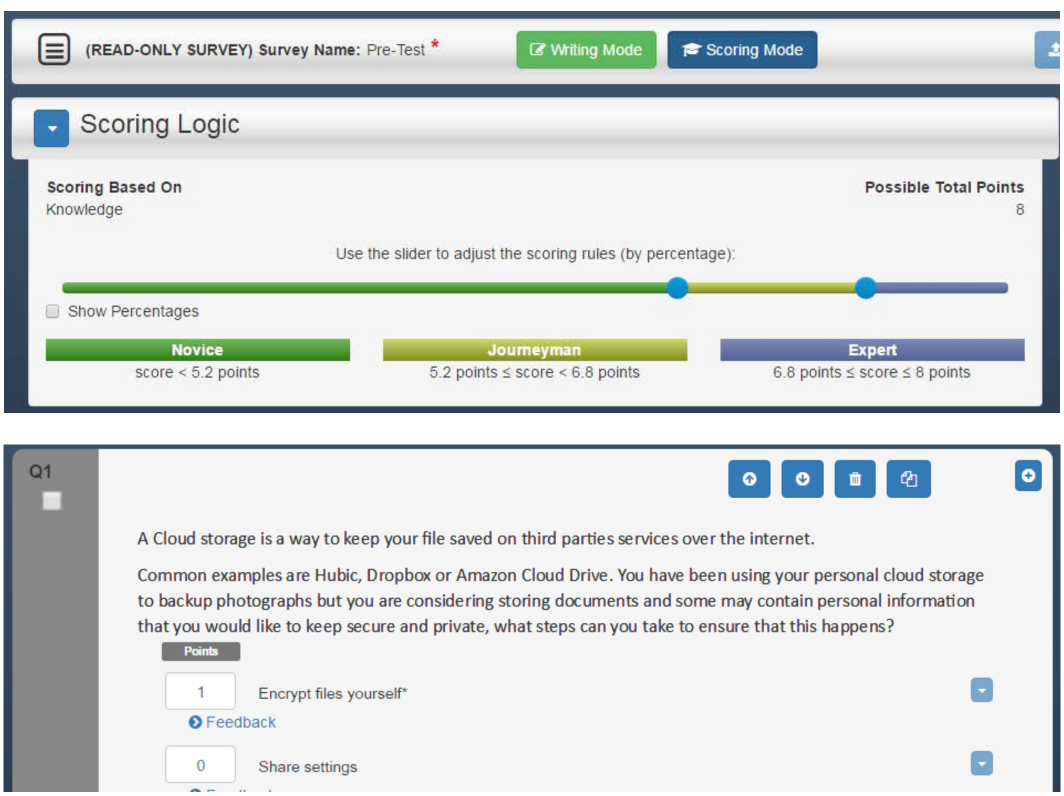


Figure 4: Survey with scoring mode activated. Scoring logic dynamically updates with score values.

Scoring logic also appears at the top of each scored survey. The design intent was to reduce the possibility of making errors in scoring a survey, by selecting UI controls that were appropriate to the task. As authors set the numbers of possible points for each question, the scoring logic automatically calculates the maximum total points. This maximum is used to set a slider for specifying the scoring logic to sort scores into three bins: *novice*, *journeyman*, and *expert*. The range of possible scoring outcomes is clearly shown on screen under each bin. Scoring can be set based on a percentage score, or raw point totals (scoring on fractional points is also possible using the sliders). The author can also set the two sliding points on top of one another to create a binary classification system, if desired (e.g., pass / fail).

Lastly, an "import from .qsf" feature is now available in the survey composer. This is an example of a function that was added in direct response to community feedback. It was recognized that some GIFT users already had experience in creating survey content using Qualtrics, a survey creation and data collection tool, used primarily for marketing research. Qualtrics exports content from their system in a .qsf format. The "import from .qsf" feature in GIFT scans the export file for survey questions and takes one of three actions: imports the question as it appeared in Qualtrics, adapts a Qualtrics question type to a GIFT question type, skips question import and notifies the author. Because the primary purposes of GIFT and Qualtrics are different, the question types supported by each are not a one-to-one mapping. Therefore, GIFT imports questions from ".qsf" files when the integrity of the question can be maintained.

Additional enhancements to the authoring tools are in various stages of development and release. We continue to collect feedback from current and potential authors in order to determine the suitability and efficiency of the tools in order to provide a positive and productive experience for authors of all skill levels. We encourage readers to examine the authoring tools further by visiting cloud.gifttutoring.org.

USER TESTING

The new user experience of the GIFT Authoring tools was largely inspired by user feedback and comparative analysis; however, there is currently little formal data available to measure and evaluate the usability, efficiency, and perception of current and future authoring experiences within GIFT Cloud. To that end, a round of formal usability evaluation was recently completed, regarding elements of the new authoring tools, described above. The objectives of this research were to evaluate the usability of the current version of the GIFT Authoring Tools (GAT) by: (1) establishing a baseline of task-performance measures, (2) establish a baseline for the overall usability of GIFT, and (3) identify potential design concerns to be addressed in order to improve the efficiency, productivity, and satisfaction with GIFT.

This section reports on a subset of the data collected during the evaluation, with a comprehensive report in preparation for publication in the coming months. Specifically, this section will describe some of the subjective, quantitative survey data collected throughout the study.

Participants

The study was conducted with twelve participants that were new to GIFT, are instructors and/or instructional designers and are familiar with military training / instruction. As this was conducted as a usability evaluation of the GIFT authoring tools, no additional demographic data was collected about the participants.

Procedure and Tasks

Participants interacted with GIFT on a laptop computer and were asked to perform eight authoring tasks (Table 1). These tasks were representative of typical authoring tasks within the GAT and were specifically designed around elements of the interface that were recently redesigned (e.g., course editor, survey system). Prior to the usability tasks, participants received training in the form of a preparatory overview of adaptive training, the GIFT platform, how GIFT enables adaptive training, and superficial information regarding GIFT's primary features and functions. Training was administered by PowerPoint and in-person lecture.

Table 1: Description of usability tasks and scenario descriptions

Task description (not given to participant)	Scenario (given to participant)
1. Add a new "Information as Text" course object to the course. Name it, "Welcome".	**1.** You have been tasked with creating some Cyber Security training content. You want to use GIFT to create a basic lesson. First, you want to display a welcome message that will be displayed to the learner when they start the course. How would you go about setting up the first screen a learner would see in a training lesson?
2. Edit the content of the "Welcome" course object. Make the text bold and blue.	**2.** You now want to edit the text of the Welcome screen to read, "Welcome to the course". You want text to be bolded and blue. How would you go about editing the text in GIFT?
3. Add two course concepts to the Course, "Internet Privacy" and "General Security".	**3.** The lesson you are creating will cover two concepts, Internet Privacy and General Security. How would you go about specifying these lesson concepts inside of GIFT?
4. Add a Media course object to the course, add an image from the desktop to the course object.	**4.** As part of the lesson materials, you want to display an image to the learner regarding protecting accounts with two-factor authentication. You have a copy of the relevant image on your computer desktop. How would you go about adding this image to your lesson?
5. Add a three item questionnaire to the course, name it "Pre-Test".	**5.** You want to add a simple pre-test survey to the lesson in order to gauge learners' existing knowledge. You have 3 sample questions contained within a notepad file on the computer. How would you go about administering the survey within GIFT?
6. Add a Slide Show course object to GIFT, use a PowerPoint show file from the desktop.	**6.** Some of your existing lesson material includes a slide show in the form of a PowerPoint show. You want to display this content to the learner after the pretest. How would you go about adding this content to GIFT in the desired order?
7. Edit the existing Adaptive Courseflow course object, by adding two web links to the Rules and Examples quadrants, respectively. Tag the content with appropriate metadata.	**7.** Your Cyber Security lesson now includes an adaptive component, made up of lesson materials and a quiz. You want to add some supporting web-links to the lesson material contained within the adaptive portion of the course. How would you go about adding this content in GIFT, and making sure it is tied to the correct lesson concepts?
8. Make a copy of the completed course.	**8.** You have recently completed the first lesson of the Cyber Security course, and are ready to get started on the second lesson. Instead of starting from scratch, you want to use the first lesson as a template. How would you go about making a copy of the first completed lesson in GIFT?

After the training, participants were presented with the list of tasks they would attempt to perform, and were asked to rate their *expectation* of how easy or difficult each task would be (using a 7-point scale). Expectation / experience ratings (Albert & Dixon, 2003) leverage a single-item questionnaire which is administered before and after a usability task, respectively:

- Before all tasks (expectation rating): "How difficult or easy to you expect this task to be?"
- After each task (experience rating): "How difficult or easy did you find this task to be?"

Those questions appeared with a seven point response scale where (1) is Very Difficult and (7) corresponds with Very Easy. The post-task ratings have been reported to be significantly correlated ($r = 0.46$, $n = 227$, $p < 0.0001$) with objective task completion rates and times (Tedesco & Tullis, 2006). Differences between expectations and experience ratings can also be used to identify opportunities within a system.

Following the completion of all usability tasks, participants were asked to complete the System Usability Scale (SUS), is an industry standard tool for quickly measuring the usability of systems and software (Brooke, 1986, 1996). The survey was designed to be administered after all tasks have been completed, but before any additional discussion or debriefing. The survey consists of 10 questions, with a 5-point response scale where 1 = strongly disagree, and 5 = strongly agree.

Preliminary analyses

For each of the eight usability tasks, the expectation and experience ratings, respectively, were averaged across all participants. The resulting pairs of values allow us to quickly gauge the health of each of the tasks, by plotting pairs of values along an X-Y axis (Figure 5). The axis is segmented into four quadrants, representing opportunities for improvement (i.e., perceived to be easy, but was difficult), features that could be promoted (i.e., perceived to be difficult, but was easy), and so on (Tullis & Albert, 2013, p. 132). For this particular usability evaluation, each of the eight evaluation tasks were, on average, expected to be relatively easy and were perceived as relatively easy once completed.

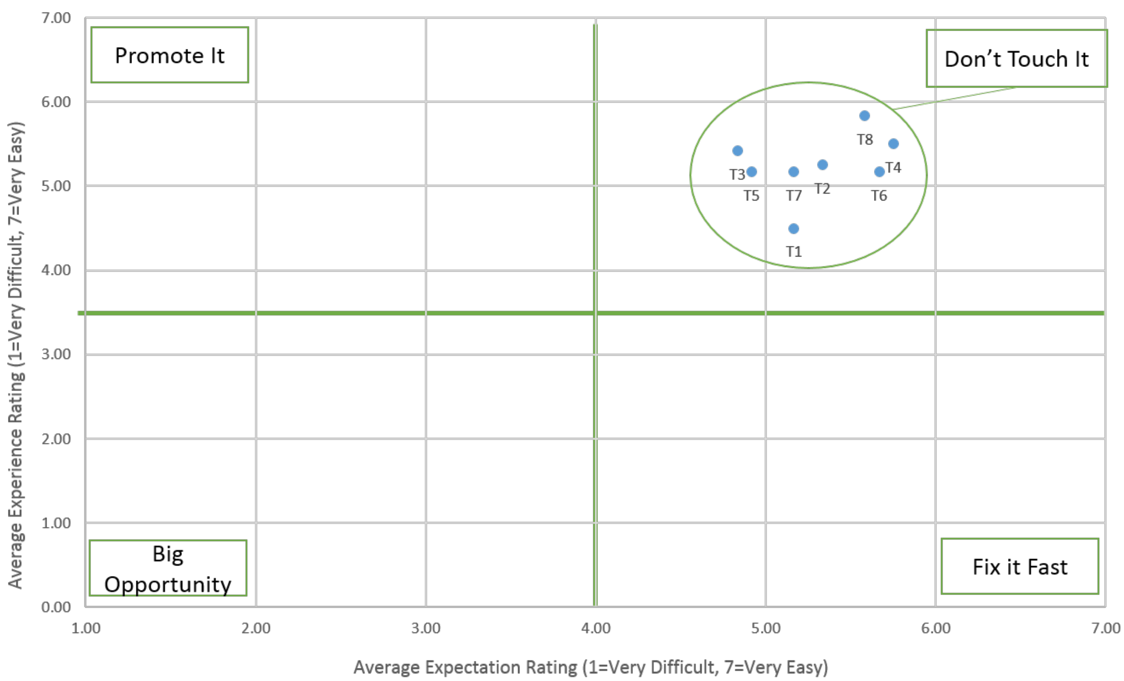


Figure 5: Average subjective expectation and experience ratings by task

The expectation and experience ratings provide a subjective snapshot of a system at the individual task level. The SUS, by comparison, provides a snapshot of the overall subjective usability of a system. The SUS score is calculated by reverse coding the even numbered questions, summing the data from all questions, and then adjusting the scale of the score to 100, however that is not a percentage score. Grades are assigned to scores based on the percentile into which the score falls (like a bell-curve). The GAT currently does not have its own SUS reference score; however, an external reference database of other system SUS scores is available for comparison (Sauro & Lewis, 2012, pp. 198-200). For the current usability study, the GIFT authoring tools yielded an average SUS score of 58.75, which ranks in the 29th percentile of all other systems in the database (roughly a D or D+ ranking). While there is room for improvement within the GAT, that result represents a modest improvement over a previous usability survey in which *perceived ease of use* for GIFT was measured (Holden & Alexander, 2015).

Preliminary discussion

The current study offers a snapshot in time regarding the subjective usability of GIFT at the system and task levels, respectively. The results reported in this paper indicates that GIFT has experienced a modest usability gain from the last usability survey, but there is room for improvement. Interestingly, subjective experiences measured at the task level were positive, which is encouraging because these were the areas of GIFT which received the most design attention. It is expected that the overall usability of the system will continue to improve as other interfaces within the GAT receive targeted design attention.

Further, there is additional objective performance (i.e., types of errors) and subjective qualitative data (i.e., task times, error rates) that is currently under analysis. There will be more conclusions to be drawn from the study, once those data have been analyzed, and opportunities for improvement can be triangulated from multiple data streams. For instance, observation of and discussion with participants identified media upload and management as an early candidate for additional design attention in the coming year. Finally, this research will also now serve as a new baseline for internal comparison with future versions of the GAT, where *expectation ratings* and the SUS are used, respectively.

LOOKING AHEAD

The new interfaces described in this paper provide usable tools and experiences that are intended to help authors configure tutors, sequence courses, and integrate instructional content into those courses. In theory, it would be desirable to have specific content relative to each adaptive component of a tutor, with respect to features within a learner model. However, *creating* instructional content suitable for adaptive tutors is still a resource intensive task, even when existing source material is available. In the coming year, the GIFT team will investigate forward-thinking solutions that will support authors in creating tailored learner experiences within a tutor, by reducing the time and effort spent in content preparation.

GIFT can already monitor and interact with external simulation and serious game systems. Technologies are being investigated that would allow single training or game scenario to be automatically reconfigured over hundreds of variations to provide suitable adaptations within GIFT. This is known as automated scenario generation (Zook et al., 2012) or evolutionary scenario generation (Luo, Yin, Cai, Zhong, & Lees, 2016). Scenario permutations would be automatically ranked by the system based on an author’s specified learning objectives. Further filtering can be done by the author and their human collaborators. This would allow a GIFT tutor to respond in real-time to a learner’s needs without minimal additional development. This capability would also benefit learners in the generation of replayable, simulation-based practice material within a course. For more information see (Sottilare & Brawner, 2017).

Further, adaptation within a tutor is commonly operationalized based on content: a remedial video, targeted feedback, more difficult quiz questions, and so on. Alternatively, is it thought that there are ways to provide an adaptive tutoring experience to learners that is, in-part, content-independent. For instance, inspiration can be drawn from gamification, or the application of game elements to non-gaming contexts. Gamification is typically implemented as a one-size-fits-all salutation, but adaptive systems afford the ability to intelligently tailor gamification-type features within the tutor-user interface that harmonize with a learner's motivation, grit, etc. (Ososky, 2015). Personalization is another way in which content-independent adaptive tutoring can be achieved. The design of personalization varies by implementation, though the premise is consistent. Personalizing learning content, with a learner's preferences or other information from their profile, can yield positive benefits on engagement and knowledge retention (A. Sinatra, 2016; A. M. Sinatra, 2015).

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

The GIFT team remains committed to providing an authoring experience for GIFT which results in greater efficiency with which tutors can be created and managed. This paper described an effort to redesign significant portions of the GAT in service of those goals. Preliminary results from a user test support the notion that the revised authoring tools were met favorably; however, there remain opportunities to improve the authoring experience as a whole. GIFT should continue to provide design attention to authoring interfaces that are found to be difficult to use and/or understand, such as media management and real-time assessment authoring.

Interfaces, however, are only part of a user's experience with a system. Future research and design effort should continue to work toward creating supporting material such as authoring *wizards*, tutorial videos, and demonstration courses that showcase the benefit of the GIFT platform (i.e., *killer apps*). In support of a comprehensive user experience, a GIFT "Summer Camp" is currently in development, which will offer in-person training directly to members of the GIFT community. While the primary goal of this offering is to support new and current authors, the GIFT team will also likely find much to learn about how individuals use GIFT as well.

Finally, the GIFT project should continue to peruse opportunities to regularly measure the usability and efficiency of the GAT. That research needs to be able to keep pace with that of software development to enable both efforts to inform one another. Future research should also leverage results from the current user study as a point of comparison in order to measure improvement, alongside the use of publically available reference databases as a general indication of usability. In doing so, data-informed decisions can be made regarding the allocation of design and development effort for the GIFT authoring experience.

As always, we encourage you, the reader, to *join the conversation* at GIFTTutoring.org. The members of the GIFT community have a valuable opportunity to help shape how features are designed and implemented into GIFT. The GIFT development team encourages members of the GIFT community to continue to communicate feedback, issues, suggestions, and results (of research) in order to help us provide the useful tools, powerful technologies, and positive user experiences that will make adaptive tutoring technology accessible and valuable to the broadest possible audience.

REFERENCES

Albert, W., & Dixon, E. (2003). Is this what you expected? The use of expectation measures in usability testing. *Proceedings of the Usability Professionals Association 2003 Conference, Scottsdale, AZ.*

Brooke, J. (1986). System usability scale (SUS): a quick-and-dirty method of system evaluation user information. *Reading, UK: Digital Equipment Co Ltd.*

Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability evaluation in industry, 189*(194), 4-7.
Holden, H., & Alexander, M. (2015). An analysis of GIFT user experiences and design recommendations for enhancing GIFT's usability. In R. Sottilare (Ed.), *Proceedings of the Second Annual GIFT Users Symposium* (pp. 83-89). Orlando, FL: U.S. Army Research Laboratory.
Lightbown, D. (2015). *Designing the User Experience of Game Development Tools*. Boca Raton, FL: CRC Press.
Luo, L., Yin, H., Cai, W., Zhong, J., & Lees, M. (2016). Design and evaluation of a data-driven scenario generation framework for game-based training. *IEEE Transactions on Computational Intelligence and AI in Games*.
Ososky, S. (2015). Opportunities and Risks for Game-Inspired Design of Adaptive Instructional Systems. In D. D. Schmorrow & M. C. Fidopiastis (Eds.), *Foundations of Augmented Cognition: 9th International Conference, AC 2015, Held as Part of HCI International 2015, Los Angeles, CA, USA, August 2-7, 2015, Proceedings* (pp. 640-651): Springer International Publishing.
Ososky, S. (2016a). Designing the user experience of the GIFT Cloud authoring tools. In R. Sottilare & S. Ososky (Eds.), *Proceedings of the 4th Annual GIFT Users Symposium* (Vol. 4, pp. 145-156). Orlando, FL: U.S. Army Research Laboratory.
Ososky, S. (2016b). Practical requirements for ITS authoring tools from a user experience perspective. In D. D. Schmorrow & M. C. Fidopiastis (Eds.), *Foundations of Augmented Cognition: Neuroergonomics and Operational Neuroscience: 10th International Conference, AC 2016, Held as Part of HCI International 2016, Toronto, ON, Canada, July 17-22, 2016, Proceedings, Part II* (pp. 55-66): Springer International Publishing.
Ososky, S., & Brawner, K. (2016). The GIFT 2016 community report. In R. A. Sottilare & S. Ososky (Eds.), *Proceedings of the 4th Annual GIFT Users Symposium* (Vol. 4, pp. 63-74). Orlando, FL: U.S. Army Research Laboratory.
Ososky, S., & Sottilare, R. A. (2016). *Heuristic evaluation of GIFT authoring tools*. (ARL-SR-0353). Orlando, FL: US Army Research Laboratory Retrieved from http://www.arl.army.mil/www/default.cfm?technical_report=7645.
Sauro, J., & Lewis, J. R. (2012). *Quantifying the user experience: Practical statistics for user research*: Elsevier.
Sinatra, A. (2016). A Review of Self-Reference and Context Personalization in Different Computer-Based Educational Domains. In R. A. Sottilare & A. Graesser (Eds.), *Design Recommendations for Intelligent Tutoring Systems: Volume 4-Domain Modeling*. Orlando, FL: U.S. Army Research Laboratory.
Sinatra, A. M. (2015). A personalized GIFT: recommendations for authoring personalization in the generalized intelligent framework for tutoring. *International Conference on Augmented Cognition*, 675-682.
Sottilare, R., & Brawner, K. (2017). *Automated Scenario Generation Broad Agency Announcement*. US Army Research Laboratory.
Tedesco, D., & Tullis, T. (2006). A comparison of methods for eliciting post-task subjective ratings in usability testing. *Usability Professionals Association (UPA), 2006*, 1-9.
Tullis, T., & Albert, W. (2013). *Measuring the user experience: collecting, analyzing, and presenting usability metrics*. Waltham, MA: Elsevier Inc.
Zook, A., Lee-Urban, S., Riedl, M. O., Holden, H. K., Sottilare, R. A., & Brawner, K. W. (2012). Automated scenario generation: toward tailored and optimized military training in virtual environments. *Proceedings of the international conference on the foundations of digital games*, 164-171.

ACKNOWLEDGMENTS

The research described herein has been sponsored by the U.S. Army Research Laboratory. The statements and opinions expressed in this article do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

ABOUT THE AUTHOR

Dr. Scott Ososky *is a Postdoctoral Fellow within the US Army Research Laboratory's Human Research and Engineering Directorate (ARL-HRED). His current research examines mental models of adaptive tutor authoring, including user experience issues related to authoring tools and interfaces. His prior work regarding mental models* *of human interaction with intelligent robotic teammates has been published in the proceedings of the Human Factors and Ergonomics Society, HCI International, and SPIE Defense & Security annual meetings. Dr. Ososky received his Ph.D. and M.S. in Modeling & Simulation, as well as a B.S. in Management Information Systems, from the University of Central Florida.*

Focused authoring for building GIFT tutors in specialized domains: a case study of psychomotor skills training

Benjamin Bell, Ph.D., Debbie Brown, Eduworks Corporation
Benjamin Goldberg, Ph.D., U.S. Army Research Laboratory

INTRODUCTION

As expressed in the Army Learning Model (ALM), psychomotor skills are foundational to many of the competencies that compose the U.S. Army's vision for 21st Century Soldier Competencies. Training psychomotor skills is being addressed in part through the use of sophisticated intelligent tutoring systems (ITS) that tailor and adapt instruction during simulations, and promising ITS investments have been made in numerous domains including marksmanship and tactical combat casualty care. However, current ITS authoring tools tend to lack generalization and are limited in scope. The process to develop ITS thus remains time-consuming and costly. For the Army to successfully realize the ALM vision, creating ITS that target psychomotor skills must be an affordable, replicable, and reusable process.

The Generalized Instructional Framework for Tutoring (GIFT) is supporting ALM by developing new tools and methods for streamlining ITS development. In this paper we report on the development of the Psychomotor Skills Training Agent-based Authoring Tool (PSTAAT), an agent-assisted ITS authoring tool for the GIFT framework. Our approach uses guided examples and the agent's encapsulated knowledge of psychomotor ITS authoring. We present an integrated approach to GIFT ITS authoring that uses performance support and agent techniques to provide informative feedback and guidance to the author during the ITS development process. We discuss how psychomotor task performance models and sensor configurations can be abstracted into reusable psychomotor profiles that both simplify and streamline the design of psychomotor activities within GIFT. Finally, we present recommendations for further generalization, enhanced reusability and portability of course components and authoring support in GIFT.

BACKGROUND

GIFT is a modular instructional architecture that provides a framework for the automation of learner modeling, domain modeling, and ITS authoring, delivery, and evaluation (e.g., Sottilare, 2012; Sottilare, Goldberg, Brawner, & Holden, 2012). GIFT remains under development and is being used to capture best practices and theories and to demonstrate adaptive pedagogical approaches and learner-centric tutoring strategies based on real-time assessment. Recent capabilities such as conversational agents and online and mobile instruction have further expanded the impact of GIFT (Sottilare, 2016).

GIFT contains numerous tools and features that support authors. Significant improvements have been made since early GIFT releases, when authoring involved writing Extensible Markup Language (XML) and understanding specific behavioral and configuration controls. Basic client-side editing tools were developed to assist with XML authoring, but knowledge of the architecture and of ITS was still required. To offer tools that are more familiar, intuitive, and supportive, the GIFT team created its first browser-based authoring tool, called the Survey Authoring System (SAS), that offered a unified authoring and preview environment with features such as integrated tool tips, and searching, sorting, filtering, and managing question banks. The subsequent browser-based GIFT Authoring Tool (GAT) provided access to authoring components in the GIFT Cloud with additional advances in usability and functionality. The

GAT also began to enforce an authoring workflow by requiring that elements such as concepts and surveys be defined prior to including them in courses (Ososky, 2016).

Based on a usability survey (Holden & Alexander, 2015), Ososky and Sottilare (2016) conducted a heuristic evaluation of GIFT authoring tools that yielded numerous recommendations for further improvement of interface consistency, user-centered design, and support materials. Many of these suggestions are being implemented in the latest version of GIFT Cloud. Now called the *Course Creator*, the primary authoring tool is accessible through the GIFT Open Virtual Campus web application (Figure 1).

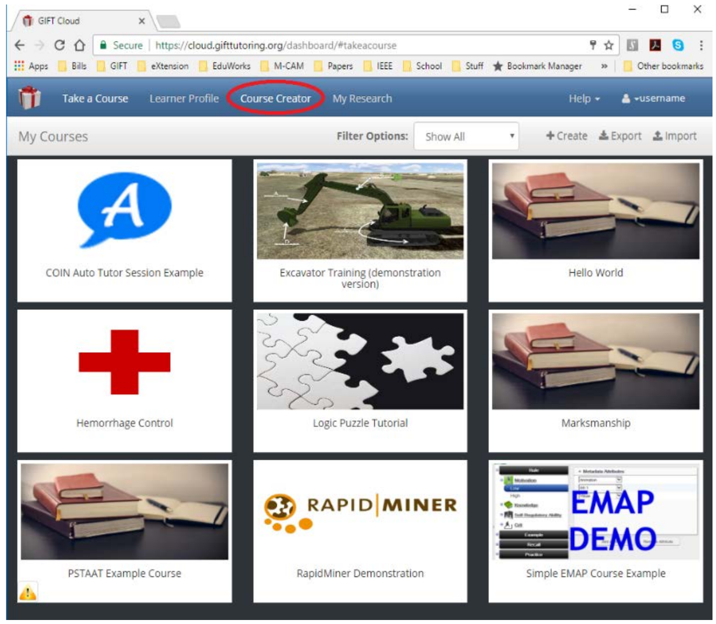


Figure 1. Course Creator Screen in GIFT

PSTAAT further extends GIFT authoring with an agent-supported tool that supports the authoring of psychomotor task training. In keeping with the direction of other improvements being made in GIFT authoring, PSTAAT provides integrated guidance informed by grounded instructional design principles and psychomotor methodologies while seeking to streamline the authoring process through workflow, templates, reuse, and semi-automation.

CHARACTERIZING PSYCHOMOTOR SKILL

Psychomotor skills can be distinguished from skills in the other domains of learning (cognitive and affective, after Bloom, Engelhart, Furst, Hill & Krathwohl, 1956). Psychomotor skills involve movement and coordination but generally de-emphasize verbal processes, and are prevalent in almost every tactical mission a Soldier might be tasked to accomplish. Tasks like fast-roping, assembling a Trident Pier, meal preparation, applying a tourniquet, flying a CH-47, aiming a weapon, or traversing a stream illustrate the prevalence and importance of psychomotor skills in performing the duties of a Soldier in today's Army.

Psychomotor skills typically include physical movement, coordination, and use of gross, fine, or combined motor-skills. The primary factor in mastering these skills is practice. Psychomotor skills tutoring should thus emphasize opportunities to practice physical skills with coaching, feedback, and assessment (Ericsson, 2006). Performance metrics for psychomotor skills are another differentiating property. Measures such as speed, precision, distance, or technique are examples of how psychomotor skill performance might be generally measured, which is a factor that tutoring systems in this domain of learning must accommodate (Goldberg, 2016).

We compared psychomotor domain models that follow the basic tenets proposed in (Bloom, Engelhart, Furst, Hill & Krathwohl, 1956) including theories advanced by Dave (1970), Simpson (1972), Harrow

(1972), and Romiszowski (1999). A simplified synthesis of these psychomotor taxonomies that appears on several university websites (Table 1) is also suitable for designing in the PSTAAT authoring tool.[2]

Each of the models represents a pedagogical progression of phases for psychomotor task instruction and is based on slightly different principles of cognitive function or learning. Selection of an appropriate model would be dependent on the type of task and requirements of the tutor. Our summary analysis established a foundation of knowledge for developing an agent to support and guide the authoring of simulation-based ITS focused on psychomotor skills, as discussed in the next section.

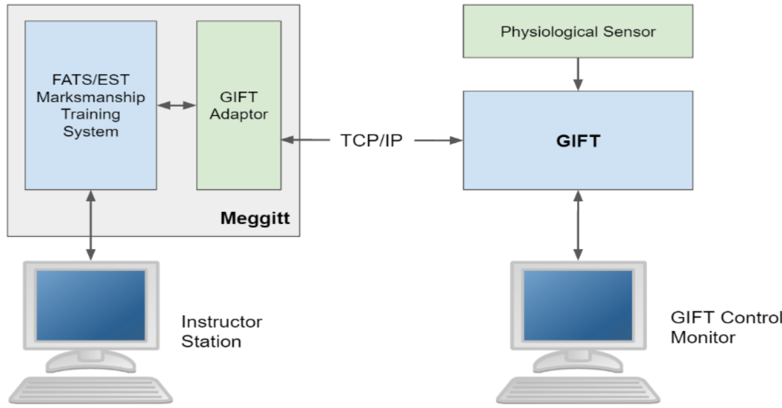


Figure 2. Overall architecture of the exemplar ITS

Table 1. Psychomotor skills domain model, synthesized from prevalent academic models.

Phase	Definition	Example
Observing	Active mental attending of a physical event.	The learner watches a more experienced person. Other mental activity, such as reading may be a part of the observation process.
Imitating	Attempted copying of a physical behavior.	The first steps in learning a skill. The learner is observed and given direction and feedback on performance. Movement is not automatic or smooth.
Practicing	Repeatedly trying a physical activity.	The skill is repeated over and over. The entire sequence is performed repeatedly. Movement is moving towards becoming automatic and smooth.
Adapting	Fine tuning. Making minor adjustments in to perfect activity.	The skill is perfected. A mentor or a coach is often needed to provide an outside perspective on how to improve or adjust as needed for the situation.

AUTHORING OF PSYCHOMOTOR SKILLS TUTORS

PSTAAT is intended to support authoring by encapsulating knowledge and assumptions about psychomotor skills training and assessment. PSTAAT uses an exemplar ITS to provide illustrations for authoring and to provide an example outcome. This approach uses an ITS built for a legacy framework and adapts it incrementally to instantiate a new ITS, a process we call *guided case adaptation* (Bell, 2003).

[2] This synthesis appears on several university websites without attribution to an original source, including Rowan University (http://users.rowan.edu/~cone/curriculum/psychomotor.htm) and Penn State University (http://archive.tlt.psu.edu/learningdesign/objectives/psychomotor.html).

Exemplar Psychomotor Skills Tutor

The Adaptive Marksmanship Trainer (AMT), our exemplar ITS, was created in GIFT to enhance an existing Engagement Skills Trainer (EST) that uses instrumented emulators of several types of firearms. AMT (Figure 2) enhances this system by incorporating adaptive tutoring and automated performance measures (Goldberg, Amburn, Brawner & Westphal, 2014). The EST makes use of a Meggitt FATS® M100 Simulation Training System and a Zephyr BioHarness to support engagement skills training. AMT processes input from 5 different sensors (breathing, barrel movement, trigger squeeze, sight picture, and shot count) to observe the learner while performing marksmanship drills. AMT uses a layered concept organization (Figure 3) to provide adaptive, contextual feedback with remedial training specific to the learner's detected performance levels for each reading (above, at, or below expectation).

Generalizing the Exemplar

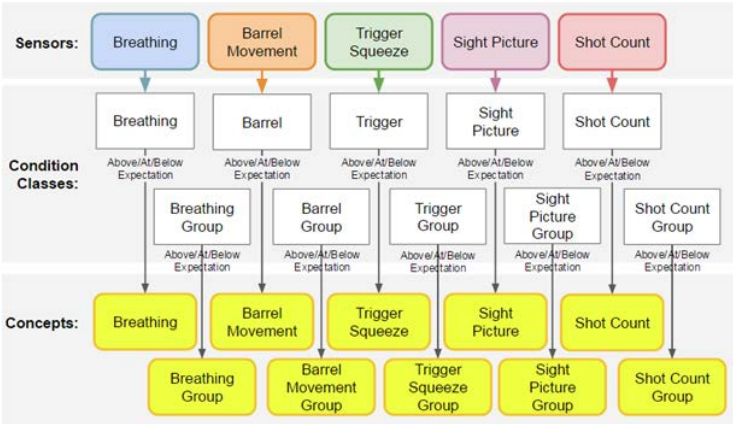


Figure 3. Basic sensor to concept mapping in the exemplar ITS

In creating a normative abstraction of the AMT authoring process, we identify steps in the workflow that can be used as the basis for a general-purpose development sequence in an authoring tool. Our workflow analysis excluded integration of hardware sensors (which is important but beyond the scope of the authoring tool). We identified three types of authoring tasks implicit in AMT: skills profiling, sensor mapping, and course object(s) definition (*i.e.*, activities and sequencing). PSTAAT provides contextual authoring support for each of these general purpose task areas, and emphasizes the use of psychomotor domain instructional approaches and adaptive feedback strategies in the form of templates and examples. The PSTAAT authoring agent is thus derived from analysis of the exemplar ITS combined with a review of ITS authoring techniques and psychomotor domain requirements. The agent initially addresses GIFT course structure and implementation; Psychomotor instructional design and strategies; Sensor model application and configuration; and Reusability and standards. Each of these is discussed below.

AUTHORING SUPPORT IN PSTAAT

PSTAAT capabilities are designed with authoring support components to be contextually incorporated into GIFT Cloud's Course Creator authoring workflow. They are envisioned as agent-driven interactions with feedback geared towards specific ITS design and development needs.

GIFT Course Structure and Implementation

To support psychomotor authoring within GIFT, PSTAAT introduces the construct of a psychomotor activity course object. A *psychomotor activity* uses configured sensors as assessment inputs and provides adaptive content delivery for associated concepts through a *psychomotor instructional approach* and the application of one or more *psychomotor profiles*. The psychomotor activity connects concept state transitions (identified by the logic model in a psychomotor profile) to instructional strategies and feedback that may be sequenced by the GIFT Domain Module (Figure 4).

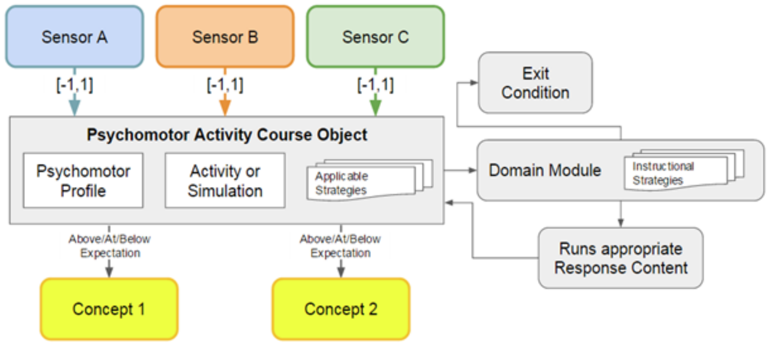


Figure 4. Example Psychomotor Course Object in PSTAAT

The authoring agent will assist during the adaptive psychomotor activity construction process by auto-generating a GIFT-compliant template for the ITS organized into phases by the selected psychomotor instructional approach and configured for the selected psychomotor profile(s) at each phase. Once the ITS template has been generated, the authoring agent will auto-populate the psychomotor activity with placeholders for all possible learner state transitions that could be detected by the configured sensors. To complete the tutor, the authoring agent will guide the author through development of instructional strategies for each placeholder state transition. For example, a particular learner state transition may reflect that the learner has performed *above* expectation in one sensor reading, but *below* in another. The authoring agent would advise that an appropriate instructional strategy for this state transition would include positive reinforcement for the skill performed above expectation along with remediation and guidance for the skill performed below expectation. PSTAAT will also assist with ITS testing by providing an activity preview feature. In preview mode, the author can step through a tutor manually selecting each possible instructional response to validate the situational appropriateness of content and feedback. When a psychomotor activity is executed in real-time, it will incorporate generalized user guidance that is appropriate for the learner's current state in the activity's pedagogical phases and instructional strategies. For example, when a learner begins adaptive remediation after performing a task, PSTAAT automatically injects appropriate messaging into the GIFT template to provide context for that transition to the learner.

Psychomotor Instructional Approach and Feedback Strategies

In the specialized case of psychomotor skills training, the tutor must have the ability to sense and observe the learner's task performance outcomes and readily adapt to any situation with appropriate instruction. In order to do this, the ITS must recognize "states" in the learner's skill performance data and respond with specific feedback and training material. Therefore, the content should be robust enough to support a wide range of previous knowledge and skill experience. For some psychomotor tasks, performance moderators such as visual acuity or physical conditioning can have significant impact on performance outcomes. In our current approach, it is possible that sensors or surveys could be leveraged to attempt to understand moderating factors and to tailor instructional feedback strategies accordingly. As part of the GIFT tutor instructional design process, the author must identify specific training concepts and associate each concept to one or more observable performance metrics provided by sensors.

Within the PSTAAT psychomotor activity, an author may choose from a set of psychomotor instructional approach templates to teach these concepts. An instructional approach template, based on a psychomotor domain model, represents a sequence of pedagogical phases that each contains appropriate activity components for all possible learner state transitions. Guidance from the authoring agent will support the author's selection of an appropriate psychomotor instructional approach for the tutor's intended behavior and outcomes. With guidance and examples from the authoring agent, the author provides response feedback strategies for all relevant learner states in each pedagogical phase by filling in activity design gaps with reinforcing or remedial content combined with contextual feedback. The instructional approach and feedback strategies are validated by the author through the use of the PSTAAT preview mode. Figure 5 depicts the authoring agent's interactions during psychomotor activity building.

Sensor Model Application and Configuration

As discussed in the previous section, a psychomotor domain ITS must have the ability to observe task performance and interpret the data collected to identify the learner's current state transition. The incorporation of sensors and the association of sensor data to performance thresholds is critical to this process. Although PSTAAT is designed to support this calibration process, it is distinct from the instructional design of the ITS. This sensor profiling is highly dependent upon the sensor hardware and the contextual authenticity in which the sensor will be used to measure psychomotor task performance.

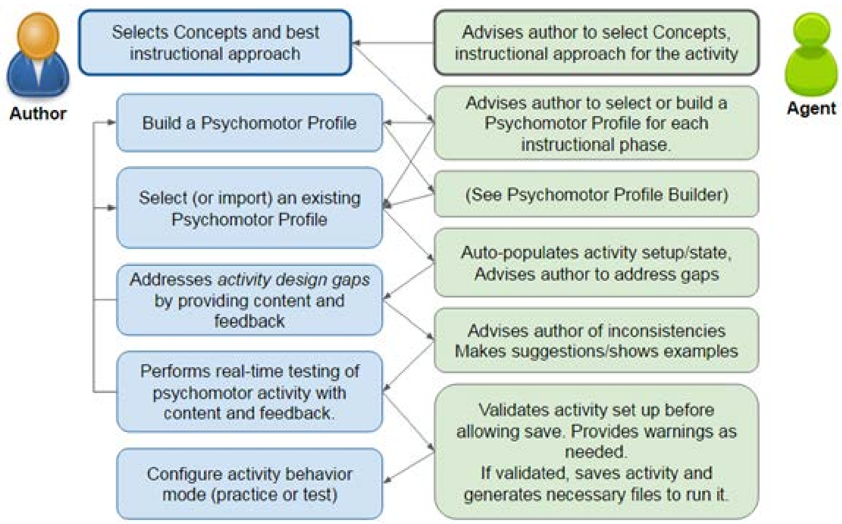


Figure 5. PSTAAT authoring agent dialog for psychomotor activity building

In PSTAAT, a *psychomotor profile* provides the logic to inform concepts based on sensor inputs. The resulting configuration of sensor inputs for each concept includes the settings for at, below, and above expected performance levels. A psychomotor profile may be selected from a set of existing profiles when designing a PSTAAT psychomotor activity. In order to encourage reuse and further streamline ITS development, psychomotor profiles may be imported, modified, exported, and reused within PSTAAT.

During psychomotor profile building, the authoring agent will implement a combination of performance support and machine learning techniques to both streamline and facilitate sensor modeling. To create the logic model for a profile, PSTAAT will require "case data" describing the sensor inputs gathered for a set of concepts at varying levels of performance. The user will have the option to import existing case data or use an integrated machine learning process to collect case data and allow the PSTAAT authoring agent to guide the modeling process. In instances when normative sensor input performance levels are known, the user will also be able to manually enter the values for the logic model (Figure 6).

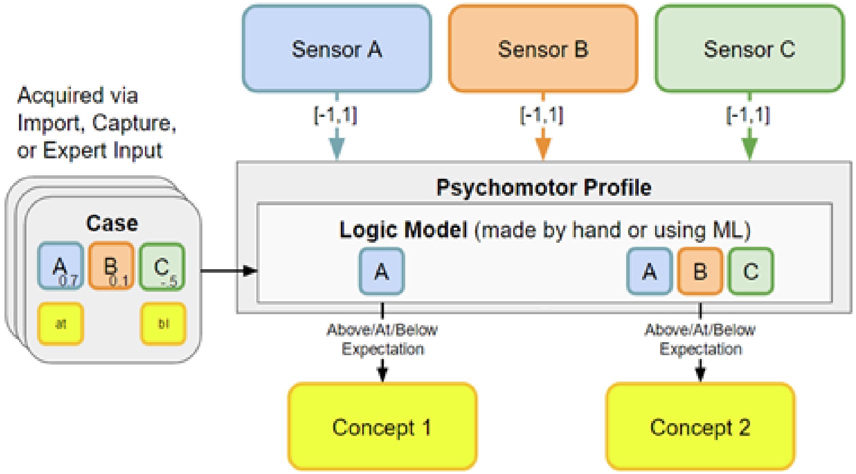


Figure 6. Example Psychomotor Profile in PSTAAT with 3 sensors and 2 concepts

PSTAAT's process for building a psychomotor profile using supervised machine learning will follow a two-step process, where the first step enables the algorithm to learn how to estimate the categorical results (i.e., below, at, or above expectation) and the second helps the author to set proper thresholds. During this process, the system will employ supervised training methods to construct one or more machine learning models that will be used to convert sensor data into concept performance levels. Throughout the sensor modeling process, the authoring agent acts as the expert coach and provides analytical feedback on the estimated model accuracy. A sample dialog with the authoring agent is depicted in Figure 7.

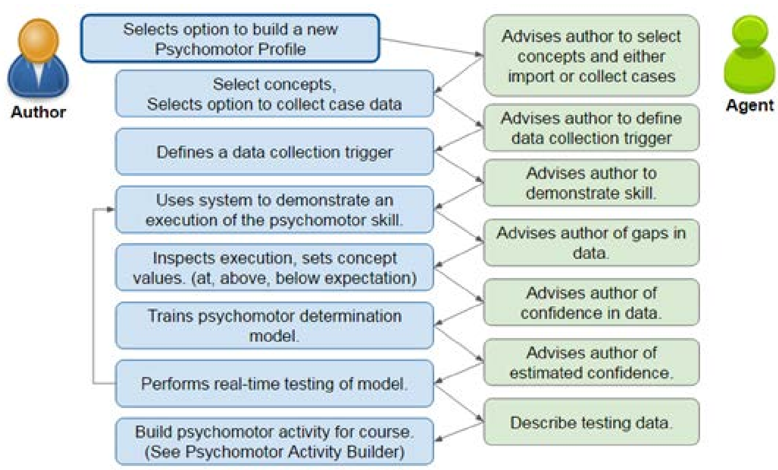


Figure 7. Authoring agent dialog to build a psychomotor profile using Machine Learning

Promoting Reusability

Several practices can improve the affordances for reuse. PSTAAT will leverage existing GIFT Cloud components and workflows, e.g., the "Adaptive Courseflow" course object in the Course Creator. PSTAAT will in addition offer reusable instructional approach templates, instructional feedback templates, and psychomotor profiles. The psychomotor activity course object provides template and profile management features for this purpose. PSTAAT will provide built-in templates, some extracted from the exemplar ITS. PSTAAT users will also have the ability to create new custom templates and profiles. Reuse is extended beyond a single course through export and import of templates and profiles.

The authoring agent can encourage the practice of reusing templates and course objects through properly sequenced guidance and the availability of convenient export/import options. The agent can provide examples and guidance on how to organize concepts to improve prospects for reuse. Similarly, when the authoring agent auto-generates portions of an ITS, the components will be assigned predictable, human-readable names that can be used to interpret pedagogical phase, concept, and sensor associations. Consistent naming conventions and the use of common web standards will ease the overhead associated with maintenance of the PSTAAT ITS as well as the interpretation of completed ITS course files.

INTEGRATION

PSTAAT will contribute to GIFT a new authoring component called a *psychomotor activity* course object, which can be dragged into a course flow like any other course object. This component will be designed to utilize existing Course Creator authoring elements and to provide a contextual authoring agent. This agent will guide the user through the creation of psychomotor activities and psychomotor profiles via a sequence of guided dialogues appearing in the Contextual Help panel (Figure 8).

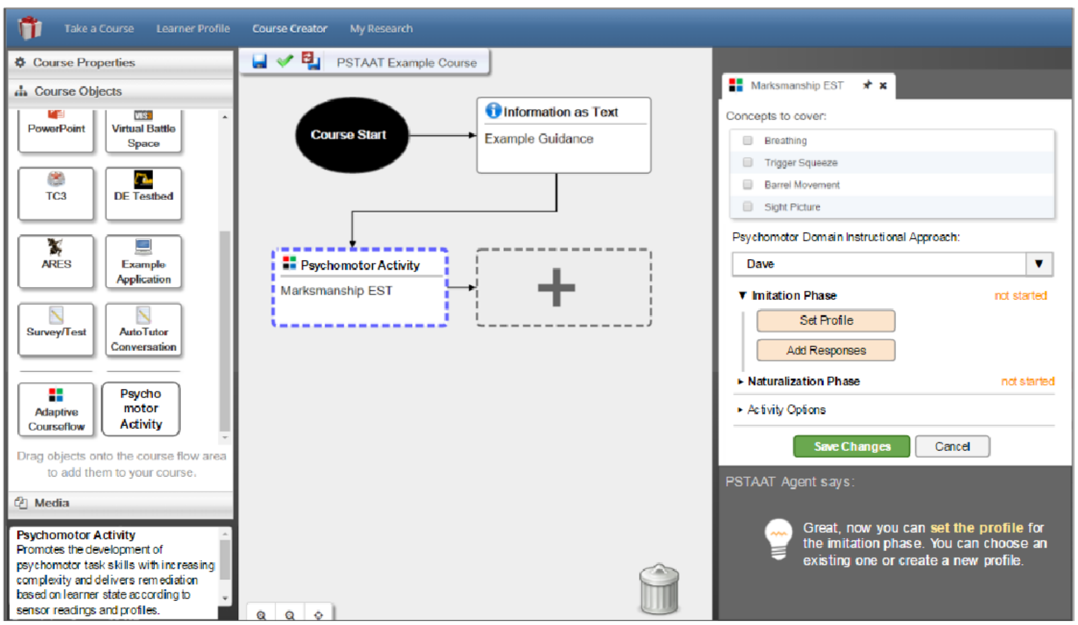


Figure 8: UI integration of the Psychomotor Activity Course Object

A psychomotor activity will be associated with a psychomotor instructional approach, one or more psychomotor profiles, and any content associated with instructional strategies to be used for remediation. Additional configuration steps may be necessary as the ITS design unfolds, as the author incorporates additional GIFT components (e.g., natural language dialogue) as part of an instructional strategy.

A psychomotor profile is composed of links to a set of sensors, a set of concepts, configuration settings related to the capture of a psychomotor event (e.g., “capture the last second when the trigger is pressed”), a psychomotor logic model that interprets captured sensor data and assigns values to concepts, and links to cases used in training the psychomotor logic model. The agent will be composed of a set of dialogue screens that, based on user input and the state of the ITS, will guide the user through the entire psychomotor activity creation process (Figure 9). The agent will be informed of the addition and use of sensors, concepts, the creation and training of a psychomotor profile, and the configuration of the psychomotor activity, including the inclusion of instructional strategies.

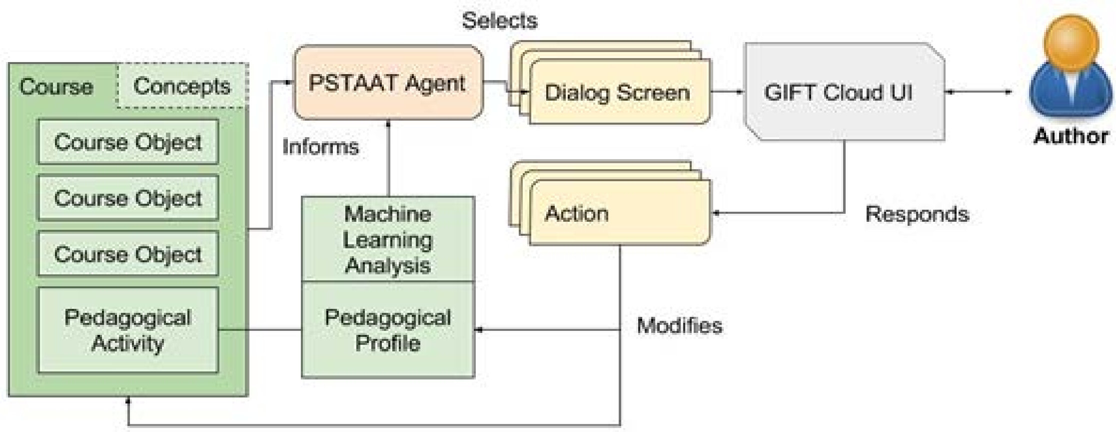


Figure 9: PSTAAT Agent Functional Block Diagram

CONCLUSIONS AND FUTURE RESEARCH

PSTAAT embodies a number of decisions made to reduce the cognitive complexities and costs associated with ITS development, to improve prospects of reuse, and to provide contextual knowledge and guidance relevant to psychomotor task instruction. Several features new to GIFT are being introduced in PSTAAT including: an activity preview feature, the abstraction of sensor configurations into reusable profiles; an agent-guided instructional design workflow; the use of templates to represent lower-level design components like psychomotor instruction and feedback; and lower-level design component management.

PSTAAT is currently addressing psychomotor skills that can be readily practiced through one or more simple sensors, or systems that can be reduced to simple sensors. Complex, sequential psychomotor skills will require more models, a wider set of sensors, and robust machine learning such as the use of Recurrent Neural Networks, Deep Learning, and other techniques that capture and interpret a wide range of input.

Currently, GIFT does not provide a generalized template management service that could be used across all of its interface objects and components. Such a service would potentially facilitate future expansion in object variety as well as user-customized objects. Similarly, since the authoring tool does not run in the the GIFT course delivery environment, native support for intelligent agents is not present while authoring. The PSTAAT team is exploring the use of a generalized performance support system. Performance support templates could encapsulate tool workflow and domain knowledge organized by user state and could be used to select appropriate feedback. Such an approach could provide a simple, common method of providing guidance that is not dependent on traditional, real-time intelligent agent architecture and services. The PSTAAT psychomotor activity course object will implement localized versions of these features, but it is our hope that the capabilities could be generalized across GIFT tools in the future.

REFERENCES

Bell, B. (2003). Supporting Educational Software Design with Knowledge-Rich Tools (2003). In T. Murray, S. Blessing, and S. Ainsworth (Eds.), *Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software.* Kluwer: Dordrecht, NL.

Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain. New York: McKay.

Dave, R.H. (1970). Psychomotor levels. In R.J. Armstrong (Ed.), *Developing and Writing Behavioral Objectives*. Tucson, Arizona: Educational Innovators Press.

Ericsson, K. A. (2006). The influence of experience and deliberate practice on the development of superior expert performance. *The Cambridge handbook of expertise and expert performance*, 683-703.

Goldberg, B., Amburn, C., Brawner, K., & Westphal, M. (2014). Developing models of expert performance for support in an adaptive marksmanship trainer. In *Proceedings of I/ITSEC*.

Goldberg, B. (2016). *Intelligent Tutoring Gets Physical: Coaching the Physical Learner by Modeling the Physical World.* Paper presented at the International Conference on Augmented Cognition, Toronto, Ontario.

Harrow, A. (1972) A Taxonomy of Psychomotor Domain: A Guide for Developing Behavioral Objectives. New York: David McKay.

Holden, H. & Alexander, M. (2015). An analysis of GIFT user experiences and design recommendations for enhancing GIFT's usability. In R. A. Sottilare (Ed.), Proceedings of the Second Annual GIFT Users Symposium (pp. 83-89). Orlando, FL: U.S. Army Research Laboratory.

Ososky, S. (2016). Designing the user experience of the GIFT Cloud authoring tools. Proceedings of the 4th Annual GIFT Users Symposium.

Ososky, S. & Sottilare, R. A. (2016). Heuristic evaluation of GIFT authoring tools. (ARL-SR-0353). Orlando, FL: US Army Research Laboratory.

Romiszowski, A. (1999). The development of physical skills: Instruction in the psychomotor domain. In Instructional-design theories and models: a new paradigm of instructional theory (Vol. 2). Mahwah, NJ: Erlbaum.

Simpson E.J. (1972). The Classification of Educational Objectives in the Psychomotor Domain. Washington, DC: Gryphon House.

Sottilare R. (2012). Considerations in the development of an ontology for a Generalized Intelligent Framework for Tutoring. International Defense and Homeland Security Simulation Workshop, in *Proc. of the I3M Conference*. Vienna, Austria, September 2012.

Sottilare R. (2016). Designing and Building Intelligent Tutoring Systems to Support Tailored Instruction for Every Learner. Seminar sponsored by the UCF Modeling & Simulation Program. Orlando, Florida, June 2016.

Sottilare, R.A., Brawner, K.W., Goldberg, B.S., & Holden, H.K. (2012). The Generalized Intelligent Framework for Tutoring (GIFT). Concept paper released as part of GIFT software documentation. Orlando, FL: U.S. Army Research Laboratory – Human Research & Engineering Directorate (ARL-HRED).

Sottilare, R.A., Goldberg, B.S., Brawner, K.W., & Holden, H.K. (2012). A Modular Framework to Support the Authoring and Assessment of Adaptive Computer-Based Tutoring Systems (CBTS). In *Proc. of Interservice/Industry Training, Simulation, and Education Conference* (I/ITSEC), Dec, 2012.

ABOUT THE AUTHORS

Benjamin Bell *is the president of Eduworks Government Solutions and an expert in the application of artificial intelligence to decision support, simulation, training, and human-machine interaction. He has practiced in the field of AI for over twenty years, leading funded research and development in applied settings, largely for military applications. He holds a Ph.D. from Northwestern University and is a graduate of the University of Pennsylvania.*

Debbie Brown *is a software engineer and senior learning technologist at Eduworks Corporation focusing on web application implementations that use AI and machine learning to semi-automate user-centered workflows and enable advanced adaptive eLearning tools and experiences. She has been conducting eLearning R&D projects for*

20 years, and operating as a software engineer in government, academic, and workforce settings for 30 years. She holds an MS in Instructional Systems and a BS in Computer Engineering from Mississippi State University.

***Benjamin Goldberg** is a member of the Learning in Intelligent Tutoring Environments (LITE) Lab at the U.S. Army Research Laboratory's (ARL) Human Research and Engineering Directorate (HRED) in Orlando, FL. He has been conducting research in intelligent tutoring for the past eight years with a focus on adaptive learning in simulation-based environments and how to leverage Artificial Intelligence tools and methods to create personalized learning experiences. Dr. Goldberg holds a Ph.D. from the University of Central Florida in Modeling & Simulation.*

GIFT Course Creator Wizard

Savannah Murray, Christina Pico, Kirsten Redmon, & Charles Rowan
United States Military Academy at West Point
Behavioral Sciences & Leadership

INTRODUCTION

The Generalized Intelligent Framework for Tutoring (GIFT) is a computer-based tutoring system (CBTS) framework aimed at increasing the ability to conduct self-regulated learning in the Army (Sottilare, 2012). GIFT can be used to author tutoring systems and CBTS components, manage instruction of selected tutoring principles, and to analyze effectiveness of the CBTS. For example, rather than sitting Soldiers in a classroom and teaching them about Combat Life Saving strategies, GIFT allows the commander to create an adaptive online course and allows the Soldiers to take this course at their own pace. In this paper, we are concerned with the authoring of an Intelligent Tutoring System (ITS) in GIFT—specifically with the potential to create a Wizard that will allow users to create a course in an expedited and user-friendly process. We will begin by providing background on GIFT's current authoring system and explain how ITSs' function. Next, we will describe heuristics accounted for and human factors components to consider. Finally, we will discuss the process of determining steps to author an ITS, present our prototype and explain how it can be implemented in GIFT to improve the system.

Literature Review

GIFT is designed to allow user feedback to dictate the system behavior, such as repeating certain learning modules if users underperform on the assessment, or prompting the user for more information regarding learner's state (Sottilare, 2012). Combining adaptive hypermedia and ITS's creates an adaptive ITS (Phobun, 2010). The adaptive ITS allows for a more personalized tutoring system for the user. Additionally, pedagogical models are important to consider when developing an adaptive ITS. Learning is a highly individualized process and, as such, should be adaptable to best fit each student. The ITS should adapt to learner ability, goals, and prior knowledge. In GIFT, this can be either set to the default setting or a customized system determined by the developer. In an adaptive ITS, there is a need for the system to collect data, analyze it, and use that data to improve the learning process (Bicans, 2015). Adaptive tutoring allows authors to individualize the student's experience to fit their individual needs.

The current authoring process in GIFT is user-centric, allowing the user complete control over the creation of their course in an affordable and simple process (Sottilare, 2012). The domain-specific knowledge in GIFT includes both authoring new elements as well as reusing previous ones—such as learning objectives, media, tasks, performance measurements, and concept maps. While this allows users freedom, it also provides users less constraints and prompts along the process than an automated process, such as a wizard, would. Among GIFT's authoring goals are decreasing the effort and skill threshold for authoring CBTS, developing and exploiting common tools and interfaces, and developing interfaces that are able to be widely used (Sottilare, 2012). Keeping these goals in mind, we created a structured interview process that allows the average, "non-programmer" to create their desired course using GIFT and create an effective learning module for users (Murray, 1999).

Heuristic evaluations are simple and helpful ways to find usability issues in a user interface design (Nielsen and Molich, 1990). Using heuristics is an efficient and affordable way to improve usability of a product. To create a user friendly product, authors must ensure that their interface is easy to learn, is accurate, productive and affordable. Using heuristics is also a cheap and affordable way for authors to test the usability of their product and receive precise and accurate feedback from users (Nielsen, 1992). When developing an ITS, it is important for users to be able to learn and memorize the interface design. Users who are able to quickly learn the interface will become more efficient thus resulting in more use of the interface (Nielsen, 1996). Nielsen's nine heuristics for user interface design stress detail focused errors within an interface. For the creation of the Wizard, we focused on four heuristics that are the most important for the ITS. First, visibility of system status, keeps the user informed on what is going on within the interface (Nielsen, 1995). The second, user control and freedom, allows the user to correct their mistakes. The third is consistency and standards; the interface should use the same language and display throughout the interface. Next, error prevention gives users clear signs that they are not performing a task correctly and the tools to fix that error. This means that irrelevant or extra information is ignored by the user. Finally, the interface should help users recognize and recover from errors. These guidelines will help us mold the ITS in GIFT into a more user friendly system.

Another use of interface design rules for the ITS system are Shneiderman's eight golden rules of interface design (Plaisant, & Shneiderman, 2010). Similar to Nielsen's nine heuristics, using Shneiderman's rules, we will be able to further evaluate and determine user productivity through these rules. When designing an interface, authors must consider the following of Shneiderman's rules: universal usability, power to navigate the interface and reduction of working-memory load. Users should be able to move within the interface at their own pace. To support locus of control, the user should have the power to navigate the interface. Reducing working memory load allows the system to fill in the blanks and to keep information in the system to prevent the user from becoming overwhelmed. These rules will help us create an easy to read and self-paced interface.

Additional heuristics emphasized in the creation of this wizard were developed by Budd specifically for website evaluation and focus more on information content than Nielsen's heuristics (Budd, 2007; Preece, 2015). Budd's first heuristics for websites is clarity—making the system concise, clear and meaningful. Next is the need to minimize unnecessary complexity and cognitive load, which removes unnecessary clutter by prioritizing important aspects with size, color, and alignment. Additionally, Budd stresses the importance of providing users with context in the form of navigation, feedback, "breadcrumb trails," or showing the steps in the process. Finally, Budd's heuristics emphasize the need to promote a positive user experience through use of attainable goals and rewards. Among these rules listed above, we see a common trend of creating an aesthetic appeal that signals users to the key areas of the interface, providing them with some level of context concerning where they are in the product, and reducing user workload. The rules most pertinent in our prototype are those centred on these ideas as well as maintaining user control while reducing workload. Taking into account Nielsen's heuristics, Budd's principles for website design, and Schneiderman's rules for interface design help guide our development of this prototype and create a usable and efficient product—the GIFT Course Creator Wizard.

METHOD

In creating the GIFT Course Creator Wizard, we first identified different methods and steps involved in creating an ITS. There are various schools of thought regarding the authoring of an ITS. Combining different approaches, we were able to synthesize our approach and determine a set of steps to follow in the creation of our interview process. While Cabada et al. (2011) breaks down the authoring process into two phases: specification of requirements and determining learner contents, Murray (1999) breaks the process into four main components: student interface, domain model, teaching model, and the student model. We synthesized these approaches to provide our general framework for authoring an adaptive ITS. Our first step was to create the structure of the course and broad concepts, to include course goals, concepts, and prerequisites. Step two was filling in the course content, such as pedagogical data, tags, media, and learning checks, utilizing the heuristics and rules described previously in this paper. Next, we mapped the concepts and tag items that relate in order to help the course flow. The fourth and final step in our process was to provide users with a preview of their system.

Murray's and Cabada's articles provided the basic framework for our methods. Murray's four components detail specifics on the creation of an ITS. In the student interface developers must consider which graphics and tutorials to use, as well as analyze usability and clarity (Murray, 1999). This includes heuristic evaluations, such as Nielsen's heuristics introduced previously. The domain model of an ITS includes curriculum knowledge, simulation models, and problem solving. The domain module is where the instructor will implement specific content tied to their course and create the structure of the system and define the goals and concepts.

In the GIFT Course Creator Wizard, populating the domain module is one of the first steps a developer completes. The order in which the steps were presented to the author were based on questions that guide the user to first input general information, such as personal information, then the author can input specific information based on the course that they want to create. We did this so that the Course Creator Wizard would flow in a logical order and the user would not get confused during the process. Pedagogical content knowledge identified through a state-based assessment completed by the learner that provides recommendations, which are then generalized and implemented in the domain module. The pedagogical knowledge options allow developers to use a predefined learning path, as described by Bicans (2015), or to create a customized approach, depending on their course needs. The curriculum knowledge category is where ITS developers account for varying levels of interest or subject importance to determine the appropriate course of action for each user. This is tracked with tags for the students' dominant learning style. We incorporated this into the GIFT Course Creator Wizard by allowing developers the option to create an initial survey to measure motivation, previous knowledge, and other factors deemed important to the developer. We also incorporated tags by creating a tag system that will couple topics together at the mapping stage and suggest an appropriate course map based on similar tag inputs throughout the process. By suggesting these tools to developers, we allow them to group their methods by topics, motivation levels, difficulties, or whichever other factors they deem important.

Cabada (2011) further describes how developers must specify links between content, consider students' different learning styles and account for that in the personalization aspects. By using various tags to map different students through appropriate methods, our Wizard will achieve this effect. Murray's simulation models involve joining components together and authoring rules and constraints concerning these junctions (Murray, 1999). In GIFT this incorporates the adaptive course flow options to map concepts together—such as the suggested mapping page and the user-controlled mapping page at the conclusion of the Course Creator Wizard. Developers can then include specific expertise—measured by learning checks and prerequisites—such as various knowledge, problem solving processes and procedural expertise (Murray, 1999). Specifying tutoring strategies and determining the range of flexibility given to the user allows adaptive tutoring systems to continue to grow and provide feedback to the system (Murray, 1999). We took into consideration that the authoring process must be considered for both flexibility—breadth and depth—and usability—learnability, productivity, fidelity, and cost—when considering tradeoffs in design.

For the creation of our Wizard, we first considered Murray's (1999) questions:

- To what degree should the author be constrained to a particular (favored) pedagogical model?
- Who are the prototypical authors who will use the system?
- What types of knowledge and skills should be modeled by the system?
- What is the source of the teaching and domain experience?

These questions helped to steer our initial interview process for ITS authors. We used these to shape our questions, creating more structured questions that would steer the development of their specific course.

We sought to make the current GIFT processing more automated through use of a wizard. Understanding that GIFT is created by the Army Research Lab and primarily will be used in military training, we focused on basic Army training courses and prerequisites to various schools. With that in mind, we looked to create a course that allowed developers—presumably commanders or school instructors—freedom to design in a similar way that they would teach the material, while still constraining them to common pedagogical strategies.

In GIFT, the typical user is not a programmer and therefore we wanted to walk them through the process more than the current system allows. In creating our questions for our Wizard we knew we wanted to guide the author from a broad idea or concept of a course towards specific goals. Then, assuming the authors have content—such as media, quizzes, or other text-based tools—we aim to match this content to sub-goals. We decided to add tags on every piece of content in order to allow authors to track their various concepts and goals as they continue to add to the course. We allow users to add learning checks, assuming they will want to test the effectiveness of their model. The mapping section is where users can control the order of display because most users may not add content in the exact order they wish to present it in. The tags allow an additional method for authors to group their concepts once they arrive at the mapping page of the wizard. Throughout our creation, we consistently put ourselves in the position of authoring an ITS and sought to answer any questions that arose when we attempted to create a course using the current GIFT system.

RESULTS

We created the prototype using Microsoft PowerPoint. Our prototype enhances the user experience by reducing the uncertainty in the course creation process. We walk the user through the steps to author an ITS and prompt them to consider many different options to add to their course—such as media, learning checks, and course mapping to fill their course, as well as providing them the option to add additional materials they feel necessary for their course. Figure 1 shows our content page of our Wizard. The tagging option ties into the adaptive courseflow option for the authors as they can later go through their concepts and tie it to specific content that will help each individual learn at their own pace. This makes it easier to upload personal information and media pertinent to the users' course development and it allows the user a visual of their progression. Following uploading the content, the user then sorts the content into one of the four quadrants—rules, examples, practice, or recall—which adds additional use for the tagging options. The user is prompted for more information until they are satisfied with their course. A user can choose to leave content blank and skip ahead, however, the first time they choose to do so the wizard will prompt them with a notice that they are leaving content blank, while allowing them to disable the reminder for the future.

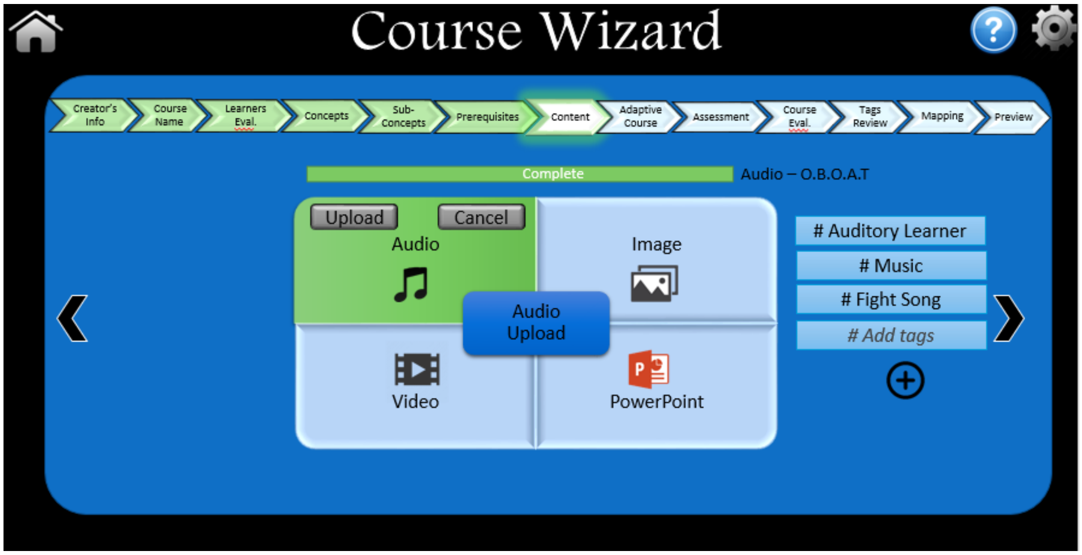


Figure 1. Content Page for Course Wizard presents the users' progress bar at the top. The tagging options are presented to the right of the various media input types. The user can upload various types of media that are used to build the personalized course.

The Wizard will help users individualize each course for its users. Much like GIFT, the GIFT Creator Wizard collects data, analyzes it, and uses that data to improve the users' learning process (Bicans, 2015). This also allows users to have control of their course without time or cost constraints (Sottilare, 2012). As previously stated, learning is highly individualized, with this Wizard users are given the freedom to add, delete or move concepts to fit their needs while using this ITS system. This ensures that users with very little experience can have success while constructing their own learning module (Murray, 1999). By reducing the process required for new users to become familiar with the system, GIFT can be more quickly learned and therefore improve the effectiveness of the system.

In creating this prototype we focused on Nielsen's heuristics of visibility of system status by adding the progress bar at the top of the page to inform the users of their progress, what they have completed, and what they have left (Nielsen, 1995). Prompting for novice users to give the current system state will facilitate the human in the loop process in order to reduce confusion and increase awareness of the tutors progress. Users maintain control and freedom with the forward and back arrows on the left and right sides of the screen that will allow them to course correct or skip any areas that they deem irrelevant to their particular course. However, we maintained error prevention, consistency and standards with the preview available at the end. The preview function presents the course authors with a user-view of the course and allows them to view the course in a partial-edit mode where they can toggle between editing and the user-view of the course.

We further enhanced user freedom by allowing authors to transfer their work to the current authoring interface once they have completed the Wizard tutorial. This will optimize user control and freedom as well as allow an additional interface in which authors can edit with more freedom once they understand the system. Maintaining visibility and system status by showing clear paths and removing clutter. We emphasized consistency in creating our Wizard. The home screen and settings screen stay consistent throughout. We also allow developers the option to choose the default pedagogical model or create a default model based on their specific goals. The algorithms of the website will run diagnostics and analysis, such as those mentioned in Bicans (2015), to provide the user feedback on their progress. We also adhered to Budd's (Preece, 2012) heuristics by designing our prototype with as little clutter as possible and using a simple layout. The simple appearance with solid colors enhances clarity for the users and provides the user with context according to Budd's heuristics. We used traditional icons for the setting and home button and placed a progress bar at the top of the page where the user can easily find it. Finally, we prompted the developer to create goals and feedback for the student to enhance student experience.

The appendix shows a complete list of questions asked throughout the Wizard to assist users along in the process. In developing the questions we used in our Wizard, we started by walking through the current authoring system on GIFT. We decided that starting broad with our questions, beginning with asking them the goal of their course and then concepts within the course, encourages authors to create and follow a structure for their course. From there, we prompted users to input specific media, texts, or tests they may have to enhance their course. We operate under the assumption that the user comes to create a course with a general idea of what they wish to accomplish and we simply aim to guide them in that process. Suggesting course evaluations—both before and after taking the course—and prerequisites for the course allows the author to consider evaluation methods to ensure they receive feedback on the effectiveness of their course as well as prompting the system as to which pedagogical method to use for the specific user. Finally, by implementing a mapping system we encourage authors to connect their ideas for the course in a manner that fits their goals.

We utilize mapping and tagging methods to enhance the adaptive courseflow. Figure 2 shows how the authors are prompted to give their standards for moving on from one concept to another. Having presented the system with tags on each piece of content lays the foundation for the adaptive courseflow to correctly draw material that a user needs additional practice on. Media and concepts tagged similarly will prompt the system to show the correct tools to users when retraining based on the learning checks. This will be a key process in the adaptive courseflow. In order for this process to work, creators must tag media, concepts, and learning check questions appropriately and map their course consistent with their goals. The algorithms and process GIFT uses in adaptive courseflow remain the same; the changes occur at the user level. They will be prompted to give specific ties between concepts and content in order to enhance the courseflow and reduce any guesswork from the process. Authors will be prompted to give specific ties between concepts, media and learning questions in order to ensure the student receives proper retraining. Once the learning questions determine a student's ability, the adaptive courseflow will determine the next step the student goes to. If they are consistently struggling with one concept, they course will automatically return them to that concepts learning content until the student is able to pass the learning questions. The threshold for moving on is set by the author and will determine when a student retrains before moving to the next concept. Additionally, authors can emplace a variety of questions and randomize the order to ensure students are not memorizing questions rather than learning concepts. The adaptive courseflow ties concepts, media and questions together and presents students with tools they need to complete the course based on their responses along the way.

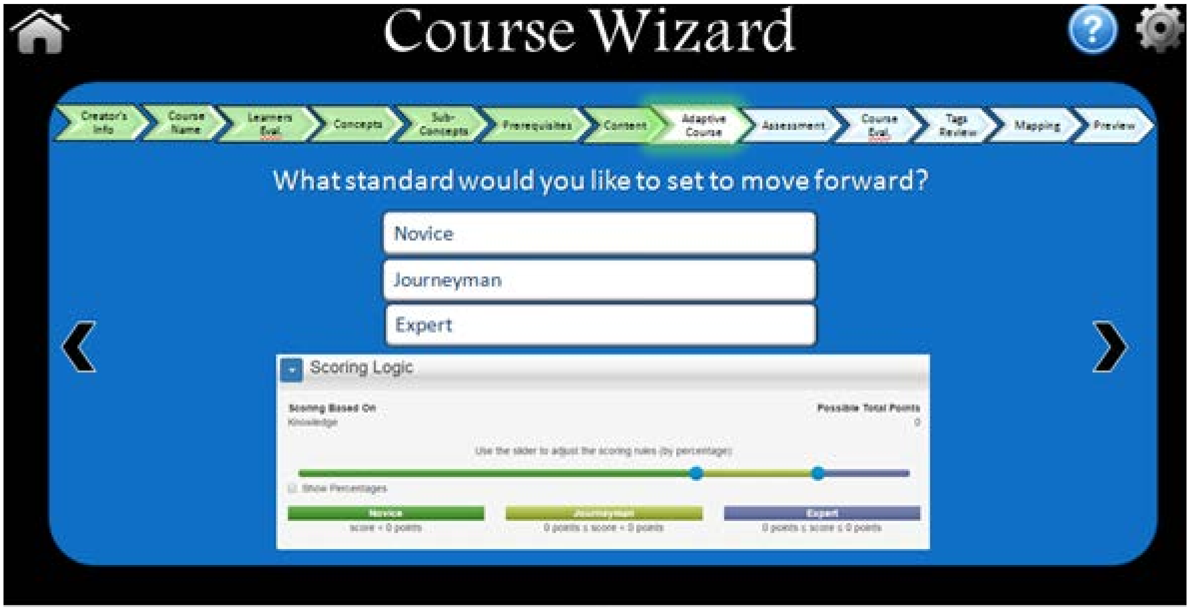


Figure 2. This screen shows the options for adaptive course flow presented to authors as they determine the standards for users to move forward in the learning module.

Transitioning from the list of questions to creating a prototype, we found arranging the questions in order to be fairly natural and intuitive—following the general order one would expect a conversation to go. By placing ourselves in the position of the authors, we found that we were able to follow a logical pattern for our questions. This prompts the user to continue moving through the Wizard. Like Bican (2015), the adaptive wizard arranges and adapts to the user's ability, goals, and prior knowledge. With this more structured process of adapting and analyzing user information, the GIFT Course Creator Wizard, keeping with the same ideas of GIFT, allows first time users to create their desired course using GIFT and create an effective learning module for users (Murray, 1999). In this process, we followed Nielsen's heuristics to design our prototype, making the steps we needed to follow to make it usable intuitive as well. In accordance with Nielsen's heuristics the wizard is based on the usability issues identified in the current GIFT system (Nielsen, 1995). This makes it easier for the users to navigate and gives the users a step-by-step introduction to the new system.

DISCUSSION

The current GIFT system allows the users freedom to create courses in whichever manner they see fit. The advantages to this include following Nielsen's heuristics of user control and freedom, and flexibility. However, the current system is complicated and has a learning curve associated with it. Additionally, the high degree of user freedom presents the tradeoff of a lack of suggestions for course development. Figure 3 shows the blank GIFT Course Creator. Rather than present users with options, the current system lays out all the capabilities and allows users to choose from the list. Users start with a blank screen with many options on the side, but there are no prompts to tell them along the way where to go from where they are. This does not guide users in developing their course and leaves room for error as they navigate the page. While the side bars present options for where to go next, the order in which the user gets there is not well structured and may require trial and error for the user to understand the process. If GIFT implemented this Wizard to present users a more structured approach to course creation, especially for military personnel designing for training purposes and predetermined course goals, the user would be able to create a more appropriate course flow and create a course that adequately trains the required material.

The versatility of the design capability of GIFT lends itself to be a functional application across various fields. This prototype can be used for application of elementary to graduate level civilian education, as well as technical schools. While the primary focus of our efforts was for use in military environments—primarily for use by commanders in situations requiring pre-training in place of traditional classroom instruction—it can easily be applied to a variety of uses. Since content drives function, application of GIFT is only as limited as the content the author uses. For example, if paramedics utilize GIFT for training, the author will choose media and content specific to paramedic training. Expanding the functionality of GIFT stresses the importance of having an interface that is intuitive and does not create a burdensome increase in time or resources in order to learn the system. This speaks to the importance of our project as we strive to bridge the gap between user expectation and performance.

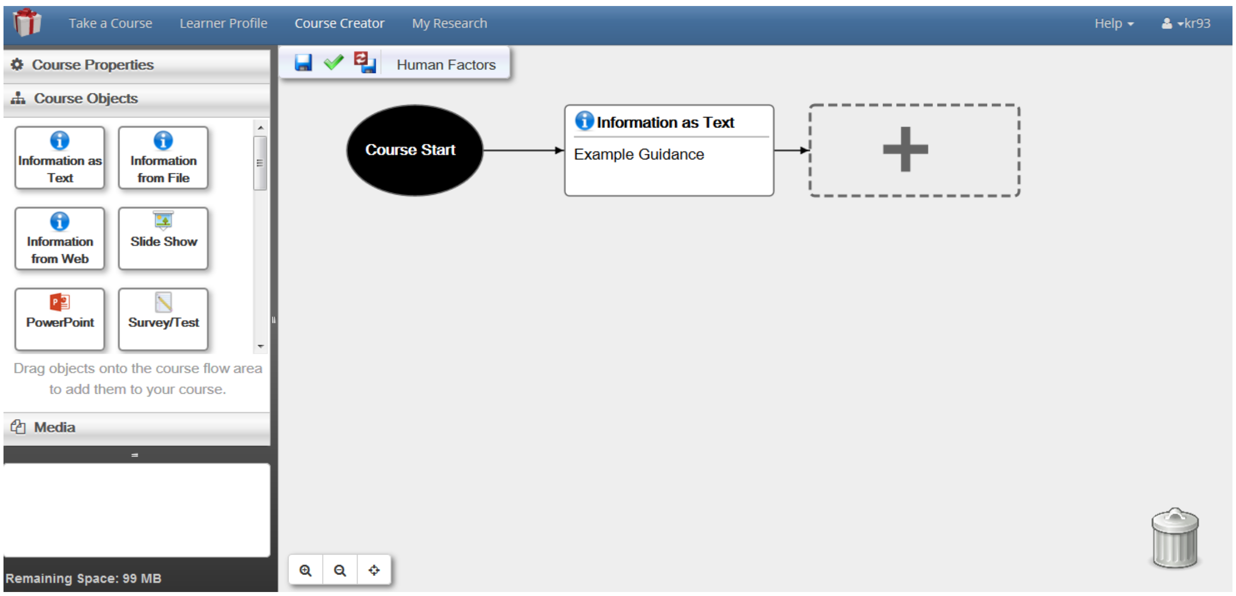


Figure 3. GIFT Course Creator for the current method. The options for users to choose from are shown on the left column as an option-bank as opposed to a prompt as the wizard would present.

The GIFT Course Creator Wizard is an easier variation of GIFT and it allows users to minimize their time spent learning how to create a course. The ability for the author to upload material as prompted, tag it with their concepts and sub-concepts, and map it as they see fit greatly reduces the user's mental workload, fitting with our heuristic suggestions. The steps necessary to uploading information and course development are easy to follow for the user and the user will be able to correct their mistakes. The general layout of the Wizard is the same as the current GIFT system. This familiarity gives the prior users more confidence when navigating the Wizard. The Wizard also implements the survey-like format of GIFT which makes navigating through the Wizard easier for the user. Mapping concepts from GIFT are also included in the Wizard, however, with the Wizard users do not map as they go, but rather create and upload as they go and then map at the end. Mapping at the end allows users to focus on each concept individually as they upload content for each concept, and then focus on the bigger picture at the end. Mapping together concepts gives the author a chance to better visualize their course and control the direction they wish to take it. Here, we adhered to Nielsen's heuristic by allowing users to maintain control of the mapping. To ease use of both the Wizard and GIFT, users can access the GIFT library and upload information through the Wizard. However, users can also upload straight to the Wizard. To make creating a course more individualized, we included a learner's evaluation—a set of questions that allow the user to identify how they learn best. This will be incorporated in the adaptive learning as it is in the current GIFT process to help steer the direction of the courses. This evaluation is another tool for users to monitor their learning style and individualize their learning process.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

Recommendations for Future Research

GIFT provides users an effective method for creating courses utilizing ITSs and allows users many advantages in their creation. However, the system could be improved with the implementation of a more structured process for users to create their courses. The GIFT Course Creator Wizard allows users the freedom to design the course the way they wish to, while also providing the structure necessary to walk users through creation and steer them towards a more student-centric design. Implementing this system in place of the current authoring system will enhance GIFT. To further enhance this system, future research should include research into the various pedagogical methods in relation to ITSs in order to make the system most effective for all users. Additionally, further developing the pre-determined learning-style questionnaire to drive the adaptive courseflow of the course would be of benefit to the authors and the students using GIFT. While this is an option now for users to create on their own, implementing this into GIFT as a standardized function and standardizing the algorithm behind it would create consistency of use. The Wizard would then be modified to present authors with different scenarios in which they would create content for highly motivated, highly unmotivated, and users in between. Creating a standard, automated system to present various types of material based on interest level would create more consistency throughout the system. The Wizard can be further developed to walk authors through by creating many different options for various learner-motivation levels.

Our prototype provides the framework for a Wizard system that synthesizes different methods in authoring an ITS. Future research should investigate the recommendations in this paper to evaluate the design and continue iterating on its structure. Specifically, we recommend usability tests consisting of creating a specific course and compare to current method between novice users and then between expert users to determine if there is an ideal method for either group. Additionally, usability testing should include participants from various populations that may use GIFT. This would include populations such as military commanders, civilian educators from elementary to graduate levels, and technical school instructors. We hope that this Wizard can be implemented and used to simplify the process of creating a course in GIFT and further developed and enhanced through the above methods.

The Wizard can be further developed to account for various learning styles, motivations, and beginning knowledge by prompting users to create content for various levels for each of these factors. This would guide authors in making their course more individualized in a simpler way as the Wizard would prompt them to account for these levels. If GIFT standardizes their learning questionnaire and levels for each motivation, the authors would then just need to be presented with the levels to fill in content. For example, the Wizard would present the author with a prompt to create content for a highly motivated visual learner and then an unmotivated auditory learner. The reason this is not in the current Wizard is because the standards for this process are not standardized from GIFT yet, making it too individualized to prompt users to create for each category. For that reason, we recommend making this change before implementing the Wizard, and then account for that change when implementing the Wizard by prompting users for each level and style and motivation when uploading content for each concept. Designing the Wizard to be more automated and consistent will enhance learnability as well provide a more enjoyable and stress-free experience for the users.

Conclusion

The GIFT Course Creator Wizard is a user friendly, self-regulated learning system that provides accurate feedback to the user. This system will adapt to the users' learning style, learning ability, prior knowledge and their individual goals for the course they are interested in. This is important because it allows the user control of their own learning while using an automatic system. To develop this system, we used heuristics for website evaluation and identified usability errors in the system and interface design. In the prototype, we focused on visibility, user freedom, consistency, and error prevention. These heuristics contribute to the users learning and productivity while using this system. The current GIFT system is not as user friendly as the Course Creator Wizard, however, if the current system is implemented, users may be more willing to continue using the course creator and it gives users a more structured advance in creating a course. The user is able to easily navigate through the Wizard by following the steps provided while monitoring their progress. The consistency of each page of the Wizard prevents confusion from the user, and expedites the process of creating a learning module. The easy to use, step-by-step process of the Wizard mitigates the possibility of errors caused by the user by confirming each step that the user completes. Using Murray's (1999) four components of authoring process and Cabada et al.'s (2011) two phases of authoring process we developed the Wizard to meet the needs of the user to provide a better base to customize and individualize a specific course. The goal of the Wizard is to allow authors to create a course with freedom and ease of use, while also presenting them with a guide to follow. GIFT course creator increases efficiency by reducing errors, reducing confusion and reducing redundancy, while allowing users to accomplish goals of ITS faster. The Course Creator Wizard encourage users to use the system and increases usability which leads to an increase in compliance rates.

When creating our Wizard, we took into consideration flexibility of system for the user, usability, learnability, efficiency, fidelity, and cost. We determined these components are most important in the creation of this user friendly Wizard. For the Wizard's course material, we created the structure of the course and broad concepts, filled in the course content, mapped the concepts and tagged items related in specific order to help the course flow, and provided users with a preview of their system. These steps allowed us to develop a personalized system that anyone can use and users will not get confused about which step they are on. In the future, the Wizard can be enhanced with a standardized adaptive courseflow that will prompt users to add content for various levels of user interest and knowledge for each learning style. This will enhance GIFT as users will have a clear idea of how to personalize their course for a variety of students. The preview allows authors to fix what they missed or are not satisfied with if the preview is not what they wish for their course. The GIFT Course Creator Wizard gives the users full access to refine user learning in a more consistent manner and can be a great asset for GIFT in the future.

The adaptive courseflow in the Wizard helps guide and retrain the user on what they know and what they do not know. This adaptive courseflow helps with efficiency and learnability. The user will be more likely to retain information that they previously needed to practice. This system can be implemented in the military to allow officers to effectively teach basic soldier skills. In the future, we hope the Wizard will be implemented in the current GIFT system and that the current GIFT system will be simplified to help with user efficiency and learnability of the GIFT system. The GIFT Course Creator Wizard is an effective way to allow users the structured freedom to create and individualize their preferred course without confusion or requiring a large amount of time spent learning the system.

REFERENCES

Bicans, J. (2015). The acquisition of context data of study process and their application in classroom and intelligent tutoring systems. *Applied Computer Systems 18*, 27-32.

Cabada, R. Z., Estrada, M. L. B., & Garcia, C. A. R. (2011). EDUCA: A web 2.0 authoring tool for developing adaptive and intelligent tutoring systems using a Kohonen network. *Expert Systems with Applications 38*, 9522-9529.

Murray, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education 10*, 98-129.

Nielsen, J., & Molich, R. (1990). Heuristic evaluation of user interfaces. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 249-256.

Nielsen, J. (1992). Finding usability problems through heuristic evaluation. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 373-380.

Nielsen, J. (1996). Usability metrics: Tracking interface improvements. *IEEE Software*, *13*(6), 12.

Nielsen, J. (1995). Ten Usability Heuristics by Jakob Nielsen. Pdf.

Plaisant, C., & Shneiderman, B. (2010). Designing the user interface. Strategies for effective human-computer interaction. *Pearson Education India, (4)*, 59-82.

Phobun, P., & Vicheanpanya, J. (2010). Adaptive intelligent tutoring systems for e-learning systems. In *Procedia: Social and Behavioral Sciences (2)*, 4064-4069.

Preece, J., Sharp, H., & Rogers, Y. (2015). *Interaction Design: Beyond Human-Computer Interaction* (4th ed.). West Sussex, UK: John Wiley & Sons.

Sottilare, R. A., Brawner, K. W., Goldberg, B. S., & Holden, H. K. (2012). The Generalized Intelligent Framework for Tutoring (GIFT). <https://gifttutoring.org/attachments/download/152/GIFTDescription_0.pdf>, retrieved January 29, 2017.

Appendix: Prototype Interview Questions

1. Create New Project or Open Project?
2. Author Name?
3. Photo for profile
4. Job Title of Author?
5. Name of Author?
6. Email of Author?
7. Contact number?
8. Gender of author?
9. Organization?
10. Course name?
11. Course image?
12. Course description?
13. Would you like to begin course with user/learner evaluation?
14. Please list course concepts & tags
15. What are the course sub-concepts (& tags)?
16. Are there prerequisites for the course?
17. Add content for concept ___
18. Would you like to use adaptive courseflow?
19. What concepts would you like to test?
20. What standard would you like to set for moving forward?
21. Create a learning assessment
22. Would you like to add end of the course learner's evaluation?

    If yes:

    a. How satisfied are you with the teaching program?
    b. How confident are you in the material of Concept ___?
    c. How satisfied are you in the course in meeting the expectation?
    d. How easy was learning the material?
    e. How motivated were you to learn the material?
    f. Overall, how satisfied are you with this GIFT course?
23. Are there any other tags to add before mapping?
24. Map the course together
25. Preview course?

ABOUT THE AUTHORS

Savannah Murray *is a senior cadet at the United States Military Academy. She is studying Engineering Psychology in the Behavioral Sciences and Leadership department. Upon graduation in May 2017 she will commission as a Second Lieutenant in the Ordnance Branch.*

Christina-Maile C. Pico *will be a graduate and Commissioned Officer in the U.S. Army as an Engineer in May 2017 from the United State Military Academy at West Point. While at the academy Pico majored in Engineering Psychology.*

Kirsten Redmon *is a senior cadet at the United States Military Academy. She is studying Engineering Psychology in the Behavioral Sciences and Leadership department. Upon graduation she will commission as a Second Lieutenant in the Engineer Branch.*

Charles Rowan *is a Major in the Simulation Operations functional area of the United States Army. He is an assistant professor in the Engineering Psychology Program in the Department of Behavioral Sciences and Leadership.*

Development of an Integrated, User-Friendly Authoring Tool for Intelligent Tutoring Systems

Fleet C. Davis[1], Jennifer M. Riley[2], and Benjamin S. Goldberg[3]
[1]Humanproof LLC, [2]Design Interactive Inc., [3]U.S. Army Research Laboratory

INTRODUCTION

The Generalized Intelligent Framework for Tutoring (GIFT) is a modular suite of capabilities aimed at overcoming the challenges associated with authoring and delivering computer-based instruction via an intelligent tutoring system (Sottilare, Brawner, Goldberg, & Holden, 2012). One of the goals for GIFT development is to create an integrated, user-friendly authoring experience that can be used across training applications. ARL, Humanproof, and Design Interactive, are currently developing the second generation GIFT Wrap prototype, a software application that allows training developers to configure the real-time, automated delivery of instructional content triggered by assessing state changes within the training application's learning environment (e.g., entity location within a virtual environment) and/or learner (e.g., progress toward concept mastery, changes in cognitive workload). This ongoing research and development effort is focused on the design and implementation of the user interface that guides users through the set-up of tutoring events. Integration with ARES (Augmented REality Sandtable) served as the first use case. ARES, which can be used as a tactical training and mission rehearsal platform, provided the context for demonstrating GIFT Wrap's utility for defining real-time assessments directly within existing training scenarios. The following sections briefly describe the previous version of GIFT Wrap, provide a detailed discussion of development efforts to date, and present concepts for extending the capability to other training applications as well as live training environments.

GIFT AUTHORING CHALLENGES

Domain Knowledge File

During the completion of a practice event within a GIFT-integrated training application, the GIFT Domain Module accesses the Domain Knowledge File (DKF) configured for that scenario. The DKF contains all of the scenario information, an ontological representation of the concepts being trained in that scenario, and parameters associated with real-time performance assessment of those concepts for pedagogical purposes. The assessment is founded upon relationships between Concepts and Tasks, which are actions completed in learning a concept. During training, the knowledge and skills of a trainee on a given Concept are evaluated using underlying Condition Classes, the rules defining how performance is scored based on available interaction data. GIFT currently uses a DKF Authoring Tool (DAT) which provides a structured, hierarchical view of Concepts and Tasks. Essentially the DAT presents to the training developer a visualization of the eXtensible Markup Language (XML) tree which houses the XML input required to execute the assessment. While users can use an XML editor to create a DKF, the DAT is designed to guide users through the creation of a DKF with the intent of ensuring the training developer creates a DKF that meets required validation criteria.

The DAT was conceptualized and created to support developers in flexibly creating adaptive training; that is, to easily support the creation of real-time assessments for a number of integrated training applications. However, the many intricacies of the system have proven too complex for the average user. For example, depending upon the complexity of the concepts and the depth of the nested hierarchy of Concepts and Tasks, establishing real-time assessments with the DAT may become difficult. This may be compounded when any number of strategies could be implemented to invoke the start of a real-time assessment or be executed following the comparison of user performance to assessment criteria. The interrelationships make comprehending the structure of the DKF and the resulting real-time assessment challenging. In fact, Shute, Ventura, Small & Goldberg (2013) described the need for users to encode conditions and instructional strategies in a DKF for real-time assessment, and highlighted the potential difficulty in understanding the structure and function of the DKF. Given the variety of training applications that can be utilized with the GIFT framework and the variety of ways performance can be assessed, users often need a high degree of skill to correctly encode Condition Classes and Strategies for the real-time assessment, as well as to create a DKF that is valid for use within the selected training application. Furthermore, as GIFT matures, more training applications may be integrated into the framework, leading to the need for substantially different conditions for performance assessment than are currently present. Currently Condition Classes are created for specific tasks via custom Java classes in the source code that are referenced by the DKF XML. GIFT currently presents any number of Condition Classes to users during the set-up of a course, but only a few may relate to the training application actually being used. If a Condition Class that is needed is not available, one must be created by building the appropriate Java classes in the source code and recompiling before building the DKF. The current DAT interface can be improved to better support users in DKF development and expand capabilities for configuring Condition Classes given functional capabilities and underlying data structures of existing and emerging GIFT training applications.

Disconnect between GIFT and Training Applications

In addition to the difficulty in creating a DKF in absence of key XML technical skills and creation of new Condition Classes for training applications, the current GIFT user must work within the GIFT course authoring application as well as any scenario or course authoring module of the selected training application to set up a course. Currently, while communication "gateways" exist between GIFT and various training applications, there is no real-time communication between the entities during the set-up of scenarios and the associated real-time assessments. As such, users must complete some set-up tasks with GIFT and then independently set-up other aspects of the scenario and or training via the training application. Currently, very limited interface support or direct integration exists that would allow a user to author a DKF while simultaneously (without cumbersome toggling) accessing or retrieving necessary data from the training application to set up task parameters, scoring conditions, or instructional strategies. Work toward supporting seamless integration for the set-up of GIFT courses would improve upon the user experience and perhaps speed up the course development process while reducing human error associated with complicated authoring functions.

AUTHORING WITH GIFT WRAP

First Generation

As previously discussed in this paper, one of the barriers to authoring in GIFT is configuring the DKF to create a set of assessments and instructional strategies for use with a particular training application. The first generation of GIFT Wrap addressed this challenge by providing capabilities for training developers to author tutoring content while simultaneously interacting with a training application's content creation tools (Hoffman, Markuck, & Goldberg, 2014). This first generation of GIFT Wrap was designed for integration with the ARES training application and provided training developers with the ability to create survey assessments of comprehension (i.e., a check on learning (COL)) including selecting specific survey response options directly from the content creation interface used by the training application (i.e., ARES terrain map). GIFT Wrap effectively provided a user-friendly tool for creating COL items and eliminated the disconnect between the two applications via its blended authoring environment (see Figure 1). This improved design significantly reduced the challenges associated with creating survey assessments that referenced scenario objects within a given training application. However, this first generation of GIFT Wrap functionality was limited to the COL. A substantial amount of additional tools and interface enhancements were needed to provide a solution encompassing the entire breadth of the ever increasing DKF functionality.

Figure 1. GIFT Wrap authoring experience showing a blended authoring environment with COL creation (left) and training application content creation (right).

Second Generation

Currently under development, the second generation of GIFT Wrap functionality aims to build upon the existing GIFT Wrap functionality by (1) providing a user-friendly tool for creating, configuring, and managing a DKF, including support for existing and future real-time assessments, and (2) further reducing the disconnect between authoring in GIFT and authoring within a training application's content creation environment. The following sections describe each of these efforts in greater detail.

User-friendly Authoring Interface

As previously stated, users often need a high degree of skill to create a DKF that is valid for use within the selected training application. The second generation of GIFT Wrap seeks to address the technical skills gap by providing an intuitive user interface (UI) for authoring a custom DKF. First, the organization of Tasks and Concepts has been restructured into a tree-like menu (see Figure 2) of "Tutoring Events".

This menu is always visible to the user allowing them to view the entire set of Tasks and Concepts at their preferred level of detail by collapsing or expanding the menu for each Task. Furthermore, the menu is purposely designed to guide the user through each step including creating a Task and then configuring the Start Trigger, set of Concepts, and finally the End Trigger. Feedback on the user's progress is provided in the form of checkmarks indicating whether or not a particular step in the process has been completed.

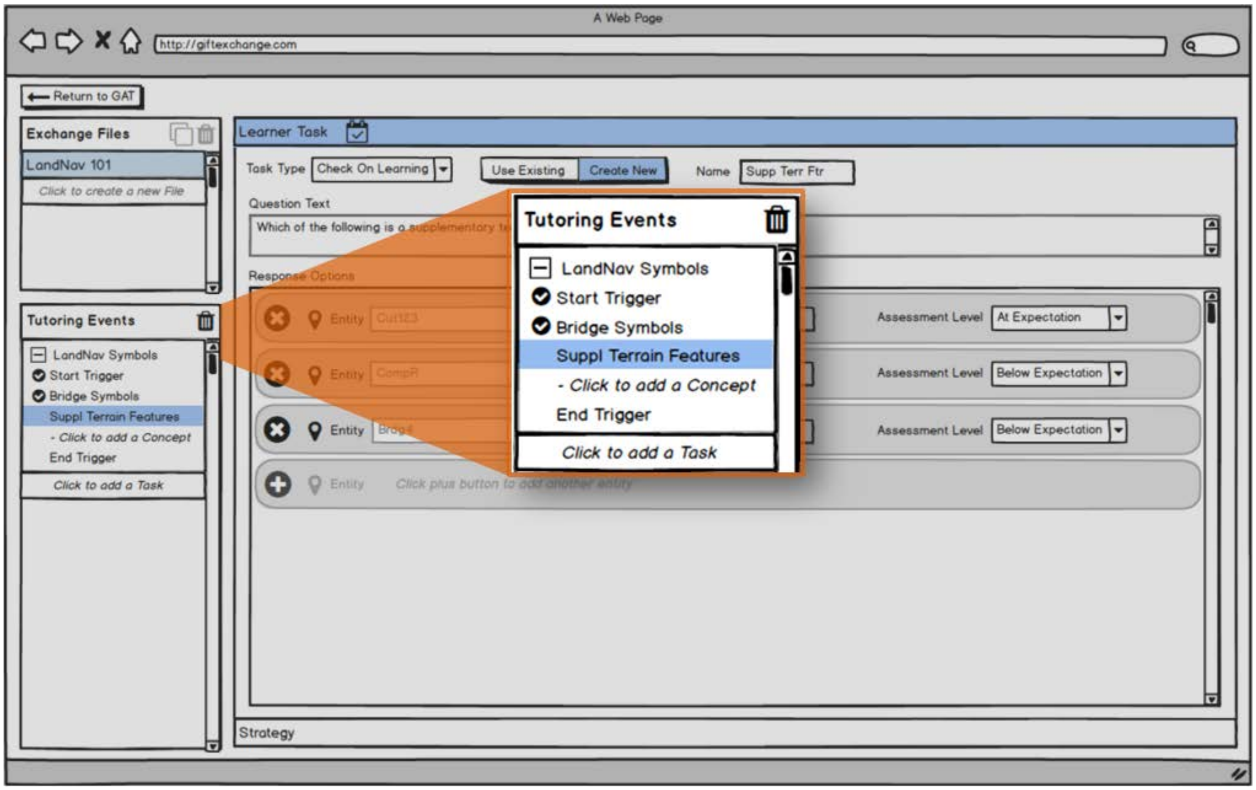


Figure 2. Tutoring Events menu guiding the user through creating a Task, configuring the Start Trigger, creating a set of Concepts, and configuring the End Trigger.

Consider the following example of how a user would interact with the Tutoring Events menu. A training developer is authoring a course on land navigation. As part of the course, the training developer would like the learner to be assessed on their mastery of map symbology using the ARES training application. Specifically, the training developer will require the learner to complete an activity that involves responding to a COL that requires them to identify two supplementary terrain features. Proceeding through each step presented by the Tutoring Events menu, they would first add and name a task "LandNav" symbols. Next, the training developer would configure a Start Trigger. In this case, they select a Timer trigger set to go off five minutes after the ARES scenario is presented. Next, the training developer would add a Concept and name it "Supplementary Terrain Features". This Concept will be assessed using a COL. After configuring the COL including the query text (i.e., Please identify two supplementary terrain features.) and response options (e.g., cut icon, fill icon), the training developer would configure an End Trigger. In this case, they select a Concept Ended trigger that will be activated once the learner completes the COL. The process is now complete unless the Training Developer would like to add additional Tasks and Concepts. While this menu is designed to support a linear process flow such as the one described in this example, it is also intended to be flexible enough to avoid hindering more advanced users.

Second, GIFT Wrap provides a redesigned interface for configuring real-time assessments and Strategies for each Concept (see Figure 3). Real-time assessments are now framed as "Learner Tasks" or activities that a learner must complete during interaction with the training application for the purpose of evaluating their mastery of a particular Concept or one of its components. Within the GIFT Wrap, each Condition Class will have its own custom-designed UI elements based on the assessment schema defined in the source code. It is also worth noting that only those Condition Classes compatible with a specific training application will be made available to the user during the authoring workflow, as it will enable a developer to understand the types of tasks and interactions they can configure their concept assessments around. For example, a new Condition Class known as a "Layout Task" has been developed specifically for use with ARES and other similar terrain associated related training applications. The Layout Task is designed to assess the degree to which a learner can place icons in the correct location on a presented two-dimensional space. In the current generation of GIFT Wrap, the task involves learners placing map icons in the correct location on an ARES presented terrain map. Again, in an effort to improve usability, the GIFT Wrap user is provided with an intuitive UI for authoring the query regarding positioning map symbology on the terrain, selecting the correct position of each element and corresponding tolerance for error, and configuring the assessment logic for each potential response from the learner. GIFT Wrap then translates the user's selections into inputs for the associated XML schema, automatically creating the DKF, in a manner that is transparent to the user.

No programming skills are required to configure the Learner Task and/or other Condition Classes available in the current generation of GIFT Wrap. Additionally, where possible, common UI elements are used for similar types of Condition Classes. For example, the COL Learner Task, which also requires the learner to select icons from the ARES terrain map, uses an almost identical UI and follows a similar process. Standardizing the UI across related or common functionality increases the usability of the tool by allowing users to quickly recognize and operate familiar functionality they've already learned to use.

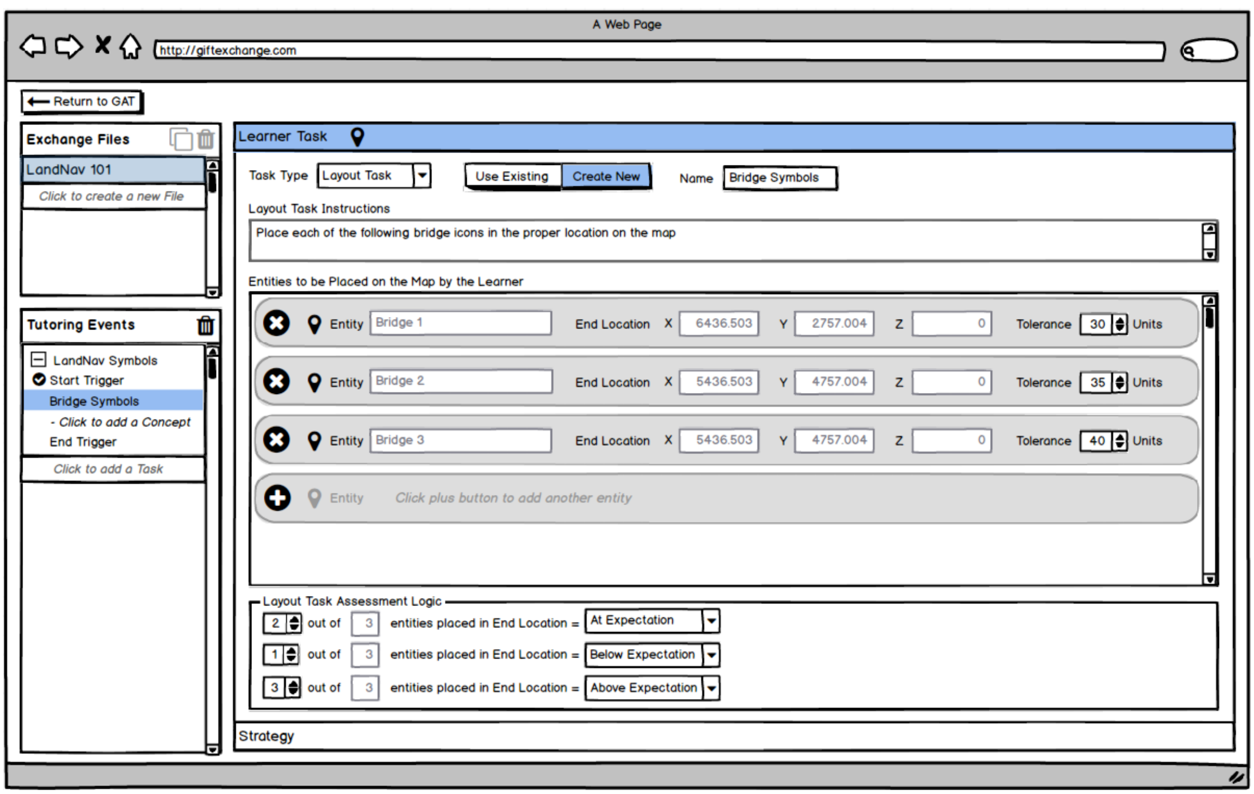


Figure 3. Flexible UI for configuring Condition Classes (Learner Tasks) and Strategies.

While the current generation of GIFT Wrap focuses on real-time assessments, including their underlying Condition Classes, relevant to ARES, future iterations of the tool will be developed that continue to incorporate the many available Condition Classes used by a DKF into the user-friendly Learner Task UI provided by GIFT Wrap. As other training applications and environments (e.g., interactive gaming environments) are explored, efforts will be made to identify new kinds of assessments, prioritize the most useful functionality for development, and to standardize functionality where possible to maintain a user-centered approach to authoring tool development.

Blended Authoring Environments

As previously stated, the first generation of GIFT Wrap established a blended authoring environment for creating COL survey assessments alongside the ARES content creation tool all within the confines of a single screen. The second generation of GIFT Wrap seeks to further overcome the disconnect between the GIFT authoring environment, in this case the DAT, and a training application's content creation tools by (1) further integrating GIFT Wrap functionality with that of the training application and (2) creating additional UI components to support the configuration of additional Condition Classes.

GIFT Wrap – Training Application Integration

In addition to COLs, GIFT Wrap now provides users with the ability to add response options for the Layout Task query by selecting map icons directly from the ARES scenario editor (see Figure 4). As demonstrated by the first generation of GIFT Wrap, this direct integration between GIFT Wrap and the training application eliminates the requirement for users to constantly toggle between the GIFT Wrap and ARES, manually entering entity IDs one-by-one. This reduces the time required to author assessments that require the user to reference an entity ID (e.g., COL, Layout Task) and decreases the likelihood for entry and selection errors thereby enhancing the usability of the tool. Furthermore, rather than being presented as a separate set of functionality from that of the DAT, the capability to create real-time assessments, such as the COL and Layout Task, is now part of the second generation GIFT Wrap interface allowing for the creation of Tasks, Concepts, real-time assessments, and Strategies all in one place.

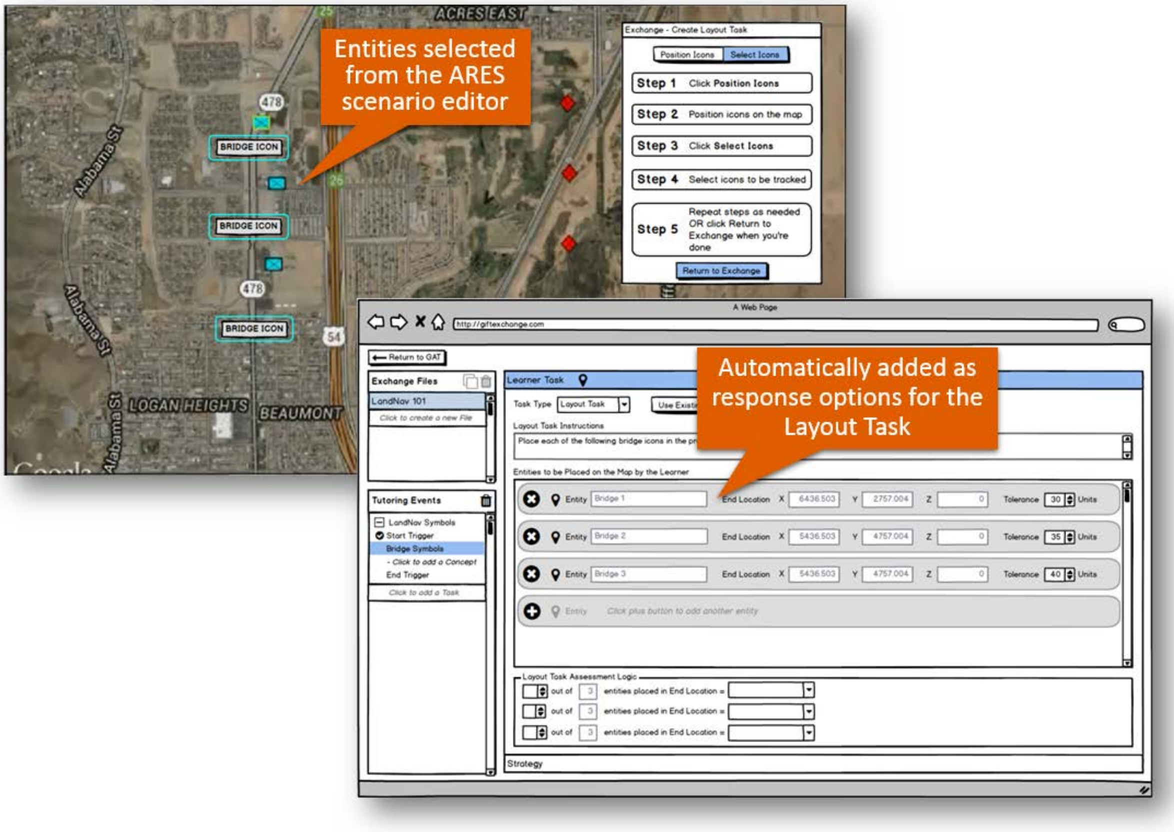


Figure 4. GIFT Wrap – Training Application Integration in support of a blended authoring experience.

Overlay UI

The first generation of GIFT Wrap presented the COL authoring UI directly alongside the ARES scenario editor within the same page (see Figure 1). While this was designed to support a one-screen solution, the GIFT Wrap UI expended a significant portion of the overall screen real estate available. In an effort to retain the one-screen design, create a framework that would support authoring additional Condition Classes, and improve the overall usability of the tool, an "overlay" UI was developed (see Figure 5). This new design includes the following benefits and features:

- The overlay is designed to be distinct from the background (i.e., a training application's content creation tools), in an effort to avoid confusion between the two environments.
- The overlay is presented on top of the underlying screen and may be repositioned as to not obscure or block access to content. It also avoids distortion or reduction of screen estate available for the training application's content creation tools.
- In an effort to guide the users through each step of authoring the Condition Class, the overlay is designed to provide a simple, visual guide (i.e., numbered steps) through the required sequence of actions. The design also allows the user to pause work with the Condition Class and easily switch to editing the training application's configuration (e.g., adjusting the location of icons on the ARES terrain map).
- The overlay presents only a sub-set of functionality (i.e., selecting icons) required to configure the Condition Class rather than all of the associated settings (e.g., query text, assessment logic) reducing the size requirement of the overlay and use of valuable screen space.

Figure 5. GIFT Wrap overlay UI supporting an enhanced, blended authoring experience.

Currently, the overlay design effectively supports configuring both the COL and Layout Task. Users can quickly alternate, or toggle, between interactions with the overlay and the ARES interface. In the near-term development of GIFT Wrap, it will be easily expandable to accommodate other map-based assessments associated with training applications such as Virtual Battlespace (VBS). In these instances, the assessment logic configured in GIFT Wrap will incorporate new data elements that will increase the complexity of measures configured in a DKF. These elements include interactions captured in first-person shooter type training applications where users control avatars in a virtual environment, and interact with scenario-based objects for the purpose of meeting task objectives. The movements and interactions across all entities and objects must be mapped for performing assessment type practices for tracking progress and identifying task deficiencies.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

Building on the first generation of GIFT Wrap, the second generation prototype successfully added a new suite of functionality and enhanced many existing features. The new GIFT Wrap UI facilitates the development of DKFs that drive real-time assessments in a guided, step-through fashion. The interface requires no programming skills (e.g., Java programming, XML familiarity) to develop a valid DKF, with associated Condition Classes and Strategies. Next, the second generation of GIFT Wrap enhanced the blended authoring experience by supporting additional capabilities, such as the Layout Task, and by providing a user-friendly overlay UI easily extendible to authoring other map-based assessments.

While the second generation of GIFT Wrap was designed to accommodate existing DKF functionality and for use with the ARES training application, the overall framework of the tool was developed with an eye toward accommodating a wide variety of future Intelligent Tutoring Systems (ITS) functionality and other training applications. In the immediate future, GIFT Wrap research and development efforts will focus on incorporating additional DKF functionality, such as new Condition Classes (e.g., follow route) and Strategies (e.g., , survey assessments), and integration with VBS. This effort will require developing new features aimed at improving and streamlining the user experience associated with creating a DKF via the GIFT Wrap UI. It will also require investigating the potential assessments that are possible to implement within VBS, identifying those most useful to training developers, and prioritizing them for implementation. These findings will guide the expansion of GIFT Wrap authoring capabilities that incorporate these additional assessments including the corresponding UI features that promote an increasingly blended authoring environment.

Following near term research and development efforts, the focus will shift toward extending GIFT-delivered training into real-world environments. Authoring within a training application's content creation tools will be replaced with authoring within representations of real-world spaces via integration with technology such as Google Maps. For example, rather than the GIFT Wrap overlay UI being presented on top of a training application's content creation tools (e.g., VBS mission editor), it will be presented on top of a real-world map. This integration with live environments will essentially create a new training application, to which all GIFT Wrap authoring functionality could apply. Inputs driving real-time assessments that were previously captured from the virtual training environment will be replaced with real-world data from trainees participating in live exercises while being monitored using Bluetooth® and other wireless technology. User-interaction with GIFT via computer displays will be replaced by leveraging mobile and augmented reality technology to interface with trainees and provide a mechanism for delivering interventions (e.g., feedback, prompting). Together with existing GIFT capabilities, this new suite of technology will allow training developers to apply many of the same ITS approaches to more traditional training environments and provide support for Army trainees as they progress from the classroom environment all the way into the field.

REFERENCES

Hoffman, M., Markuck, C., & Goldberg, B. (2016). Using GIFT Wrap to Author Domain Assessment Models with Native Training Applications. In Proceedings of the Fourth Annual GIFT Users Symposium (GIFTSym4).

Shute, V., Ventura, M., Small, M., & Goldberg, B. (2013). Modeling Student Competencies in Video Games Using Stealth Assessment. In R. Sottilare, A. Graesser, X. Hu & H. Holden (Eds.), *Design Recommendations for Intelligent Tutoring Systems, Volume 1: Learner Modeling* (pp. 141-152).

Sottilare, R.A., Brawner, K.W., Goldberg, B.S. & Holden, H.K. (2012). The Generalized Intelligent Framework for Tutoring (GIFT). Concept paper released as part of GIFT software documentation. Orlando, FL: U.S. Army Research Laboratory – Human Research & Engineering Directorate (ARL-HRED). Retrieved from: https://gifttutoring.org/attachments/152/GIFTDescription_0.pdf

ABOUT THE AUTHORS

Mr. Fleet Davis *is a Senior Human Factors Engineer at Humanproof, LLC. He is the Principal Investigator for the GIFT Wrap project. Fleet has over 15 years of experience in human factors engineering and specializes in bringing a user-centered approach to the design and development of adaptive training tools for use across live and virtual environments. Prior to joining Humanproof, he served as a researcher at SA Technologies, Inc. where he developed training and assessment tools for a variety of DoD customers including the US Army Research Institute for the Behavioral and Social Sciences.*

Dr. Jennifer Riley *is the Director of Defense Solutions at Design Interactive, Inc. She is the Co-Principal investigator for the GIFT Wrap project. She has over 15 years of experience in human factors, cognitive engineering research, and human-systems integration, with a focus on cognitive skills training and human performance measurement. Jennifer's most recent projects include the development of technologies for cognitive skills* *instruction for military applications, the development of tools translating lessons learned into reusable learning objects, and the design and development of an augmented reality system for command and control in robot-assisted military operations.*

Dr. Benjamin Goldberg *is an adaptive training scientist at the Army Research Laboratory's Human Research and Engineering Directorate. He leads research focused on instructional management within ARL's Learning in Intelligent Tutoring Environments (LITE) Lab and is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT).*

Theme III: Individual Learner Modeling

Learner Models in the Generalized Intelligent Framework for Tutoring: Current Work and Future Directions

Gregory A. Goodwin
U.S. Army Research Laboratory – Human Research and Engineering Directorate

INTRODUCTION

The function of an intelligent tutoring system (ITS) is to adapt or tailor training to an individual learner. As with a human tutor, this requires the ITS to have some "knowledge" of the learner (i.e., a learner model). The ITS uses and updates the learner model as the learner progresses through the material. For example, if the learner masters some concept, the learner model must be updated to reflect this. On the other hand if the learner has difficulty with a concept, the ITS needs to be able to understand where deficiencies lie in order to prescribe the appropriate remediation.

Understanding why the learner might have had difficulty with a particular concept is no simple task as the list of reasons could be quite extensive. Perhaps the learner lost focus during the presentation of a key piece of information, lacks some key prerequisite knowledge, or has a low aptitude for the domain. The list could go on and on.

All of these possible explanations require assessment of the learner. As can be seen from the above example, assessments can include information about the learner's background, experiences, traits, and aptitudes, as well as measures of the learner's affect, behavior, and performance during the training session. The more completely the learner model represents the learner, the better the ITS will be able to effectively adapt training.

Dimensions of Learner Modeling

In September of 2015, we published a report outlining research challenges in the area of individual learner modeling (Goodwin, Johnston, Sottilare, Brawner, Sinatra, Graesser, 2015). This report described a framework for assessment of the learner to support learner modeling. This framework provides a way of classifying different types of measures and relates those measures to adaptive methods.

The framework categorizes measures into four groups in a 2 x 2 matrix. One axis in the matrix divides measures into state-like or trait-like categories. Trait like measures are what the learner brings to the training event. Examples would include physical strength and aptitude. State-like measures on the other hand are things resulting from the training. Examples include fatigue or confusion. State-like measures are fairly stable and either don't change, or change very slowly. Trait-like measures change fairly quickly and are often transient.

The other axis in the matrix divides measures into content-dependent or content-independent categories. Content dependent categories are learner measures that are directly relevant to the content being trained. Examples include prior knowledge or comprehension. Content independent measures are traits and states that are relevant to training generally rather than to specific content. Examples include aptitude and personality traits. Each of these four cells apply to three domains of learning (cognitive, affective, and psychomotor, *vis*. Bloom, 1956).

State-like and trait-like measures have some interdependencies (Goodwin, Murphy, & Hruska, 2015). For example, a student with high aptitude or prior experience would be expected to perform better in training (Schafer & Dyer, 2013). Additionally, some state-like measures could update trait-like measures. For example, as the learner completes a block of training, his or her performance (state-like measures) would then update the trait-like measures, (e.g., indicating the learner had mastered a particular skill or completed a certification course).

ITSs need both state like and trait like measures to adapt training effectively (VanLehn, 2006). For example, before an ITS can initiate training, it needs to know something about the learner. What does the learner already know? What is the learner's aptitude? How motivated is the learner to complete the training? The ITS might use this information to determine the difficulty level of the training or what topics to skip. These are often described as outer-loop adaptation. As the ITS delivers training, it will measure student comprehension, attention, as well as the types of errors made, and level of frustration and/or boredom. The ITS can use these measures to choose remedial content or to change the pace or difficulty of the training – so called inner loop adaptation (VanLehn, 2006). Table 1 summarizes the kinds of measures that can be used for adaptation of training in GIFT.

Table 1. Components of the Learner Model.

	Learner Measure Category	**Trait-Like (Outer Loop Adaptation)**	**State-Like (Inner Loop Adaptation)**
Content Dependent	**Cognitive**	Relevant prior cognitive experience/knowledge/training	Comprehension of concepts presented in the training
	Psychomotor	Relevant prior psychomotor experience or training,	Measures of Skill improvement
	Affective	Fears, likes, goals, attitudes relevant to the training.	Arousal and emotions in response to the training
Content Independent	**Cognitive**	Intellect/Aptitude, Memory, Meta-cognitive skills	Attention, Cognitive Workload
	Psychomotor	Physical strength, stamina, sensory acuity	Endurance and fatigue
	Affective	Personality Traits, general test anxiety	Arousal, emotions resulting from factors independent of training

Using this assessment framework for developing learner models has a couple of benefits. First of all, by understanding that there are different uses for each type of assessment, it is possible to think about ways that those uses might be standardized in GIFT modules. This might be especially true for content-independent measures. Second, it is useful in identifying research and technical challenges that affect certain types of assessments.

For example, in-training assessments of learner state are challenging because they must be frequently and rapidly assessed in a nonobtrusive way by the training system. Such assessments rely on measurement technologies like eye-trackers and physiological measures that can be expensive and may only be available in certain training facilities. This highlights the need for research and development to bring the cost of these capabilities down and to increase their validity.

Assessment of trait like factors is time consuming and so we want to avoid doing this every time a learner starts a training session. Ideally GIFT would access pre-existing databases containing that information (e.g., personnel records, learner records). Research is needed to develop ways to access that information in a secure way using open standards. Services also need to be developed to facilitate interoperability among databases. The next section outlines ongoing research in the area of learner modeling.

AREAS OF RESEARCH ON INDIVIDUAL LEARNER MODELS FOR GIFT

The following are areas of research on individual learner models for GIFT that are currently being investigated:

Modeling Learner Competencies

We know that ITSs can be expected to operate within a larger ecosystem of training events and systems. For example, for a given skill or course, a learner may receive training in a live or distributed classroom led by a live instructor, participate in hands-on training, virtual simulation training, multimedia training, and/or game-based training. Often these separate events are developed and sequenced so that the learner's skill or expertise progresses throughout the course. The ITS may only deliver a single block of instruction within the larger course or may be used to provide remedial training. Both of these circumstances indicate that there is a need for a learner model that tracks learner competencies as they develop across multiple training venues and that can be shared among multiple training systems.

Competencies are domain specific knowledge and skills possessed by the learner. Competencies can encompass a large set of skills acquired over a long time (e.g., being a researcher or a physician) or they can be very specific (e.g., launching a Raven unmanned aerial vehicle). The challenge is that there are no standard, broadly accepted, validated ways to assess most competencies. Competencies are reflected not only in the training the learner has received, but also by their experience and performance of that competency in battlefield conditions. Competencies change over time though gradually. They may increase if the learner practices the competency regularly but they can decline in the absence of practice.

Because there is no standard set of assessments for most competencies, and because competencies are not static, there is a need to be able to determine competencies at the time of training. An effort (Engine for Quantifying User Intelligence and Performance – EQUIP, see Goodwin, Murphy, & Hruska, 2015) is investigating an approach to provide this capability to GIFT. The components of the EQUIP architecture include a Learner Record Store (LRS) that contains performance data relevant to the competency being assessed in an experience application programming interface (xAPI) format; an interoperable learner competency model (ICM), and of course the GIFT application.

Figures 1 and 2 illustrate this system. Let's suppose that a course deliver in GIFT were to tailor training to a learner based on the learner's current competency in some domain. As shown in Figure 1 below, the course concepts are read from the domain knowledge file (DKF) and are then passed through the gateway module to a web service that hosts a set of ICMs. The web service queries all ICMs to determine which ones may be relevant to the concepts of the course. Each ICM contains an index of performance measures and methods for interpreting those measures which are returned to GIFT.

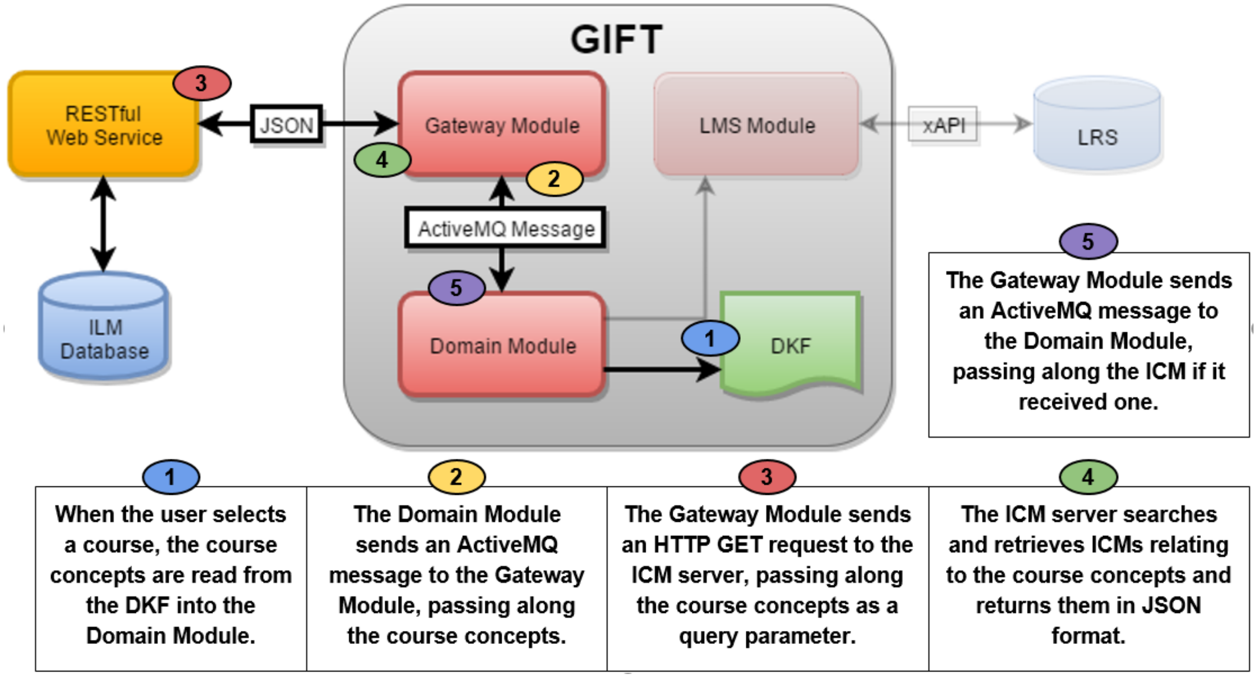


Figure 1. GIFT Integrated Architecture Flow: Steps 1 - 5

Once GIFT has the ICM data, it can then query the LRS for the appropriate performance data for that learner and subsequently interpret that data to estimate the learner's current competency level as illustrated in Figure 2. GIFT can also add assessments to the LRS. To estimate the competency level, it is necessary to have validated models to predict them.

For example, suppose we were to develop an ICM for marksmanship. The Army presently scores marksmanship competency/proficiency in four categories based on the number of hits in a standard course of fire:

1. Expert (38-40 hits; max = 40)

2. Sharpshooter (33-37 hits)

3. Marksman (26-32 hits)

4. Unqualified (25 or fewer hits)

In an initial entry training environment, students complete the marksmanship qualification test at the end of training. In that training environment, the ICM could use learner measures to make predictions about competency using the Army standard. However, it would probably also be useful to be able to use learner measures to make predictions about performance in intermediate training events. For instance, an ICM might map performance in the simulator to predictions about performance during the subsequent period of live instruction.

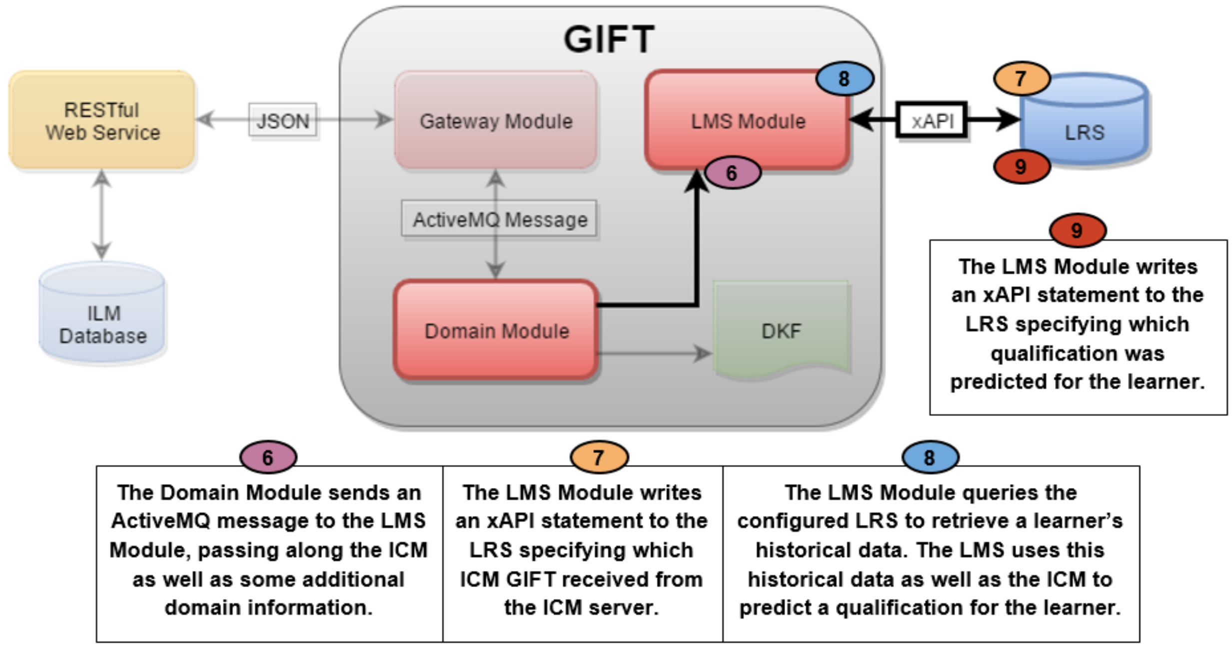


Figure 2. GIFT Integrated Architecture Flow: Steps 6 – 9

As the learner completes training in GIFT, those learner behaviors and assessments would be fed back into the LRS. That new performance data would then impact subsequent assessments of learner competencies.

The primary advantage of ICMs is that they allow for standardization of competency modeling across different training systems. Furthermore, they allow for GIFT to know more about its learners than what they have done in GIFT applications. By opening up this window to GIFT, it can much more efficiently target training to learners. In this way, GIFT can act much more like a human tutor would, as an adjunct to a course for example. Clearly, this allows GIFT to operate within a larger ecosystem of training systems including live, virtual, constructive, and gaming in a seamless way.

Increasingly, these assessments are being written using an industry standard known as the xAPI specification. This standard was developed by the Advanced Distributed (ADL) Co-Lab as a means of logging learner activities across a wide variety of platforms, systems, and media. Each xAPI statement includes a subject, verb, and object and contextual information (ADL, 2013). The specification also includes data transfer methods for the storage and retrieval of these statements from a learner record store (LRS) and security methods for the exchange of these statements between trusted sources.

Currently, data pertaining to learner actions, states, and accomplishments stored using the xAPI specification provide the best means of creating and updating a persistent interoperable learner model. In order to do this, GIFT and other adaptive training systems will need to both consume and generate xAPI statements of learner assessments that can be used to update competencies in a learner model.

Assessing Differences in Motivation: Long Term Learner Modeling

Another effort underway in the learner modeling domain involves an examination of the ways in which motivation affects the rate of learning and forgetting of a given learning task. The approach taken is to develop and validate a motivator taxonomy that matches motivators to personality traits of learners. For example, it may be that individuals who score high on measures of extraversion are most strongly motivated by acknowledgement from peers or higher ups. On the other hand, an introvert may be more motivated by free time and relaxation.

This is a three-phased project and work is currently in the first phase. The first phase focuses on the development of a Motivator Assessment based on individual differences. The motivator assessment identifies motivation in the learner. It builds upon efforts to incorporate additional classification variables that include student personality, learning performance history, and motivational responses. Motivational responses refer to a measured increase of sustained effort, because of the end goal resulting in a reward based on personality. Sustained effort would be indicated by physiological measures, such as a higher amount of oxygen produced for a longer sustained time or an increase in heart rate due to stress/arousal to meet the goal.

The second phase of this project will involve an experimental verification of the Motivator Taxonomy and/or the Motivator Assessment. Specifically, this will test how personality and the Motivator Taxonomy/Assessment affects the learning rate and retention of training. The learning objective could be presented in a simulation-enabled mission command, intelligence, surveillance, and reconnaissance mission, UMedic, or some other application to be determined. The goal for this phase, is to identify the relation between the classifications of motivational tools and individual factors with the learning rate and retention, specifically the Long Term Learner Model

In the final phase of this project, data collected from the previous scenario will be tested across a different domain, population, and/or scenario. All results will then be used as the basis for a framework that will provide pedagogical recommendations based on the evaluation of the Student's real-time data on motivation and personality factors into a specific learning intervention for GIFT training.

Modeling the Determinants of Training Time in GIFT

Adaptive training promises more effective training by tailoring content to each individual insuring that it is neither too difficult nor too easy. Another, less discussed benefit of adaptive training, is improved training efficiency. This efficiency comes from minimizing the presentation of unnecessary material to learners. Typically, non-adaptive training is developed for the lowest tier of learners. While this insures that no learner will be unable to complete the training, it also means that many students are given material that is not well suited to their current level of understanding.

The focus of this effort (Goodwin, Kim, Niehaus, 2017) is to determine how the fit between learner characteristics (e.g., aptitude, reading ability, prior knowledge), learning methods employed by the adaptive training system, course content (e.g., difficulty and length, adaptability), and test characteristics (e.g., difficulty, number of items) all determine the time to train for a population of learners.

We use a probabilistic model to represent the different factors and instructional strategies that impact the completion time of a MAST module, as well as probabilistic inference techniques to determine a distribution of a course completion time.

For example, if a trainee normally reads at 100 words per minute, there are 100 words in the text, and the trainee is tired, the reading time of the trainee could be distribution uniformly from 1 to 2 minutes. The reading speed of the trainee is also a non-deterministic variable that depends on how much prior knowledge the trainee possesses about statistics about how fast the general population of trainees read.

One of the benefits of building a probabilistic model to represent the completion time is that not all of the information in the model is needed to estimate the completion time. For example, if we know how much prior knowledge the user has about the subject (for example, from a pre-instruction questionnaire), we can post that knowledge as *evidence* to the model that would be taken into account when estimating the completion time. If we do not possess that information, we can treat the variable as *latent* and use a prior distribution to represent the state of the variable. For example, we can estimate that only 20% of trainees taking the course have prior knowledge of the subject. These prior distributions can be estimated from the literature review or expert knowledge, and then *learned* over time based on the outcomes of actual testing.

RESEARCH CHALLENGES

To date, the research into how best to adapt training content based on student performance in intelligent tutoring systems is inconclusive (Durlach & Ray, 2011). As can be seen, GIFT-based research on learner modeling is still relatively nascent. Some key areas of research that need to be investigated are described below.

Cross platform training. The major benefit of interoperable student models is the ability to adapt training across technology platforms. Using the xAPI specification, performance data can be recorded and interpreted from a wide variety of platforms, including desktop and mobile devices. While some Army-sponsored efforts have focused on assessing student performance across a range of training platforms (e.g., Spain, et al., 2013), maintaining a complex student model across these platforms – and adapting training accordingly – has yet to be successfully accomplished in a military context. Integrating GIFT with xAPI data would enable investigations into the best practices for adapting training across platforms.

Macro- versus micro-adaptive interventions. Multi-faceted student models based on cognitive, psychomotor, and affective components are inherently complex, and may be representative of both “state” and situationally dependent components such as level of workload and “trait,” or more persistent student characteristics such as personality traits. Whether to adapt training on a macro level (e.g. course selection) or a micro level (e.g. real time adaptation of content) based on these complex models has yet to be fully investigated. While some research suggests macro-adaptive strategies are more appropriate for more persistent characteristics (Park & Lee, 2004), this question has not been addressed across domains.

Adaptation based on a combination of learner states. Assessing a learner’s affective state during the course of training has been a focus of ITS research over the past decade (e.g., D’Mello & Graesser, 2007). However, research into how to adapt training based on this state is in its infancy (e.g., Strain & D’Mello, 2015). Arguably the state of the art in intelligent tutors, Affective AutoTutor (D’Mello & Graesser, 2007), senses student cognitive and emotional states such as boredom and frustration and acts to alleviate states. If a negative emotion is detected, the avatar within the tutor responds with an encouraging phrase and facial expression. In Affective AutoTutor, student affect and learning are managed through separate models; that is, interventions that are geared toward managing frustration are distinct from interventions aimed at manipulating content difficulty. The extent to which different interventions could be used to address combinations of these states has yet to be determined, but is a research question GIFT could support.

Scenario-based training. GIFT is unique in that it supports intelligent tutoring in scenario-based platforms such as the Army’s *Virtual Battlespace 3* (VBS3). How to assess competencies across complex student models using key events within one of these scenarios has yet to be investigated. If scenario data were recorded in xAPI specification scenario events could be diagnostic of both performance and affect. Key to this development is the careful mapping of competencies to decision events in a scenario. Best practices for accomplishing this have yet to be established.

Predictive analysis of performance. Persistent learner models provide the opportunity to prescribe interventions based not only on performance during training but also prior to training on both the macro- and micro-adaptive level. Based on performance in one training setting, a student model could reflect a number of cognitive, psychomotor, and affective attributes which could then predict performance in another setting, given the domains were sufficiently interrelated. These data could be used to prescribe courses of instruction, training platforms, and even micro-adaptive strategies. To date, this potential has not been investigated.

Return on investment of different types of interventions. To date, research into addressing interventions based on complex student models is feasible. However, whether or not a learning intervention is effective is not that same issue as whether or not it is effective *enough*. With defense budgets becoming increasingly limited, the question is whether adapting training based on complex representations of student competency is worth the investment. Implementing intelligent tutoring systems to date has been limited due to their domain specificity and cost to develop. While the GIFT initiative aims to address these issues specifically, the relative cost of some interventions has yet to be determined. For example, emerging physiological technology enables the unobtrusive measurement of student cognitive and affective state (Murphy et al, 2014), but does adapting training based on these types of measures produce sufficient learning gains to warrant their cost? These questions have yet to be fully investigated.

CONCLUSIONS

This discussion highlights a number of research questions that can be addressed as the result of integration of complex, interoperable learner models into the GIFT architecture. Through the use of xAPI data, representations of student performance can incorporate data from a multitude of sources. The GIFT team envisions a multi-faceted learned model consisting of psychomotor, cognitive and affective aspects of competencies. This model can be used to drive training adaptations across technological platforms, across domains, and across the course of a learner’s career. While the potential to fully model the lifelong learning of a student is promising, research is needed to fully evaluate the utility of these learner models. Some of this work is currently underway at the Advanced Distributed Laboratory under a program known as the Total Learning Architecture (TLA, Johnson, 2013).

As an initial attempt at addressing these issues, several projects are using a marksmanship use case for an initial investigations of this capability. Marksmanship is an ideal domain for implementing multi-faceted learner models. While marksmanship skills may appear to be straightforward, effective performance is much more than simply hitting a target with a bullet. The marksman must master a range of psychomotor, cognitive, and affective skills in order to be successful, and must have an understanding of how myriad environmental factors play into his or her accuracy. Furthermore, marksmanship is a skill that every Soldier must master, so it has a broad applicability to the Army and its sister services.

It is important to note research in learner modeling is still in its infancy. Consequently, our efforts are a first step toward developing definitive guidelines and best practices for how to best leverage interoperable performance data. Further research will be needed to expand an understanding of how these learner models play into the development and use of intelligent tutors across domains, training audiences, and platforms.

REFERENCES

Advanced Distributed Co-Laboratories (2013). xAPI-1.0.2. Retrieved from https://github.com/adlnet/xAPI-Spec/releases/tag/xAPI-1.0.2.

Bloom, B.S. (Ed.). Engelhart, M.D., Furst, E.J., Hill, W.H., Krathwohl, D.R. (1956). Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain. New York: David McKay Co Inc.

D'Mello, S. & Graesser, A. (2007). Mind and Body: Dialogue and Posture for Affect Detection in Learning Environments. In R. Luckin, K. Koedinger & J. Greer (Eds.), Proceedings of the 13th International Conference on Artificial Intelligence in Education (AIED 2007) (pp 161-168). Amsterdam, Netherlands: IOS Press.

Durlach, P. J. & Ray, J. M. (2011). Designing adaptive instructional environments: Insights from empirical evidence (ARI Technical Report 1297). Arlington, VA: U.S. Army Research Institute

Goodwin, G.A., Kim, J.W., Niehaus, J. (2017). Modeling Training Efficiency and Return on Investment for Adaptive Training, Presented at the 5th *Annual GIFT Users Symposium* 10-11 May, 2017, Orlando, FL

Goodwin, G.A., Murphy, J.S., & Hruska, M. (2015). Developing Persistent Interoperable Learner Models in GIFT. Presented at the *3rd Annual GIFT Users Symposium* 17-18 June, 2015, Orlando, FL

Goodwin, G., Johnston, J., Sottilare, R, Brawner, K, Sinatra, A., & Graesser, A. (2015). Individual Learner and Team Modeling for Adaptive Training and Education in Support of the US Army Learning Model: Research Outline. ARL Special Report 0336. U.S. Army Research Laboratory.

Johnson, A. (2013). The Training and Learning Architecture: Meeting the Needs of the Next Generation of SCORM. Slides from a Webinar. Downloaded from: http://adlnet.gov/wp-content/uploads/2013/02/TLAWebinarFeb2013-1.pdf on 4 Jan, 2017.

Murphy, J.S., Carroll, M.B., Champney, R.K., & Padron, C.K. (June, 2014). Investigating the Role of Physiological Measurement in Intelligent Tutoring. Paper presented at the GIFT Users's Symposium, Pittsburgh, PA

Park, O., & Lee, J. (2004). Adaptive instructional systems. In D.H. Jonassen (ed.), *Handbook of*

Research on Educational Communications and Technology 2nd edition (pp. 651-684).

Mahwah, NJ: Lawrence Erlbaum.

Schafer, P., & Dyer, J (2013). Defining Tailored Training Approaches for Army Institutional Training. (ARI Research Report 1965). U.S. Army Research Institute for the Social and Behavioral Sciences. Fort Belvoir, VA.

Strain, A., & D'Mello, S. K. (2015). Affect regulation during learning: The enhancing effect of cognitive reappraisal, *Applied Cognitive Psychology*, *29*: 1–19.

Spain, R., Mulvaney, R., Cummings, P., Barnieu J., Hyland, J., Lodato, M., & Zoileck, C. (2013). Enhancing Soldier-centered learning with emerging training technologies and integrated assessments. *Interservice and Industry Training and Simulation and Education Conference*, Orlando., FL.

VanLehn, K. (2011). The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist*, *46* (4), 197–221.

ABOUT THE AUTHOR

Gregory Goodwin *is a senior research scientist and acting Branch Chief at the Army Research Laboratory-Human Research and Engineering Directorate, Simulation and Training Technology Center (STTC) in Orlando, Florida. His research focuses on methods and tools to maximize the effectiveness of training technologies. After completing his Ph.D. at the State University of New York at Binghamton in 1994, Dr. Goodwin spent three years in a post-doctoral fellowship at the Columbia University College of Physicians and Surgeons followed by a year as a research associate at Duke University Medical Center before joining the faculty at Skidmore College. In 2005, Dr. Goodwin left academia and began working at the Army Research Institute (ARI) field unit at Fort Benning Georgia and six years later, he came to the ARI field unit in Orlando, FL where he has been examining ways to leverage technologies to reduce the cost and improve the effectiveness of training.*

Multi-level User Modeling in GIFT to Support Complex Learning Tasks

Ramkumar Rajendran[1], Naveeduddin Mohammed[1], Gautam Biswas[1], Benjamin S. Goldberg[2], Robert A. Sottilare[2]
[1]Institute for Software Integrated Systems, Department of Electrical Engineering and Computer Science, Vanderbilt University, 1025 16th Avenue South, Nashville, TN 37212, U.S.A.
[2]U.S. Army Research Laboratory – Human Research and Engineering Directorate, Simulation and Training Technology Center, 12423 Research Parkway, Orlando, FL 32826, U.S.A.

INTRODUCTION

Open-ended computer-based learning environments (OELEs) are user-centered. They present users with complex problems to solve, and a set of tools and resources that support the problem-solving task. While problem solving, users typically explore multiple solution approaches, and assess their evolving solutions to make sure they are making progress toward their learning and problem solving goals. In general, OELEs focus on developing users' (1) cognitive skills, (2) metacognitive processes and (3) problem solving strategies that go beyond the acquisition of domain-specific cognitive skills (Hannafin et al., 1994). These environments make high cognitive demands on users, and promote the development of strategies and metacognitive processes that can support planning, monitoring, and self-evaluation processes.

Novices often have difficulties in making progress when working in OELEs. To help such users with personalized and adaptive feedback, our goal is to create more detailed user models in the Generalized Intelligent Framework for Tutoring (GIFT) system (Sottilare et al., 2012), to support analysis of users' cognitive, strategic, and metacognitive processes as they work on learning or training scenarios. Previous approaches for providing feedback to users primarily focused on their performance when applying cognitive skills to solve a problem (Lester et al., 2014; Kulik & Fletcher, 2016). On the other hand, our approach dives deeper into the users' intentions when they work on the system. To interpret users' actions in context, we have developed a multi-level task modeling approach that specifies the cognitive skills and processes that helps us interpret user competence and behaviors as they work on their learning and problem solving tasks (Kinnebrew et al., 2016).

In this paper, we apply our user modeling framework to an OELE called UrbanSim (McAlinden et al., 2008), a turn-based simulation environment, where the trainee plays the role of a commander directing counterinsurgency (COIN) efforts in a Middle Eastern region. U.S. Army manuals (Nagl et al., 2005) discuss the strategic and operational implementation of COIN operations as including three phases: Clear, Hold, and Build (CHB). The idea is to deal with insurgents and empower Host Nation (HN) security and capacity building in service of the local population. In the UrbanSim environment, the officer trainee, acting on the Brigade commander's intent, analyzes the current state of the region, and performs activities directed to defeating the insurgents, while increasing the level of population support and facilitating self-governance and economic independence for the area. Trainees have access to political, economic, military, and infrastructure information about different regions in the area of operations (AO). More specifically, this also includes intelligence reports on individuals and groups, information on the stability of each region, and economic, military and political ties among their leaders.

The user model for the UrbanSim environment in GIFT is developed by considering the users' (1) proficiency in relevant domain tasks, and (2) their learning behaviors, i.e., the approaches they adopt in selecting, executing, and sequencing their tasks to achieve their learning and problem solving goals. Users' activities and actions on the system are collected from log files generated by the UrbanSim system, and interpreted by the log parser for analysis in the GIFT environment (Segedy et al., 2015). The proficiency of users in domain-specific task is measured by their ability to operationalize the mission goals, which are described by six Lines of Effort (LOE) measures. The six LOEs corresponding to the primary goals of most counterinsurgency operations are (1) Civil Security, (2) Governance, (3) Host Nation Security Forces, (4) Essential Services, (5) Information Operations, and (6) Economics. Related performance measures that can be extracted from the UrbanSim log files include: (1) Population Support, (2) Political, military, economy, social, information and infrastructure (PMESII) values of each region in the AO, (3) Military Power (MP) of the insurgents groups and the HN security force, (4) Coalition Support (CS) of individuals and tribes in the regions, (5) Effectiveness (EF) of utilities such as water storage unit, sewage treatment plant and trash depot in supporting the population, (6) Capacity (CA), that is the state of operations of infrastructures such as airport, hospitals and school.

In addition, we also analyze and interpret a set of strategies that we believe will further help us characterize the users' problem solving and decision making approaches, and their performance in the UrbanSim environment. The ability of the users to strategically implement the CHB doctrine is UrbanSim specific, but the other strategies: Situational Awareness (SA), Tradeoff Analysis (TA) and Second and Third order Effects (STE) of actions, are more general, and very likely apply to other problem solving and decision making scenarios. In GIFT, we compute users' proficiency in each strategy by using their links to the relevant UrbanSim activities and actions, as well as users' relevant performance measures (see last paragraph) provided by the UrbanSim system. The UrbanSim user model was developed by performing task analysis, consulting knowledgeable experts, and by analyzing users' interactions with UrbanSim from research studies that we conducted in past two years. In this paper, we will present our approach and demonstrate the effectiveness of the user-modeling scheme in providing instructional feedback to the users.

PROPOSED USER MODEL IN GIFT

Our proposed user-modeling framework implemented in the GIFT framework is illustrated in Figure 1. We have developed a three-tier hierarchical user model for monitoring and capturing users' cognitive skills, problem-solving strategies, and their metacognitive processes within the GIFT tutoring framework. The bottom layer of our user model derives information from logs of user activities collected from the training environment (e.g., UrbanSim).

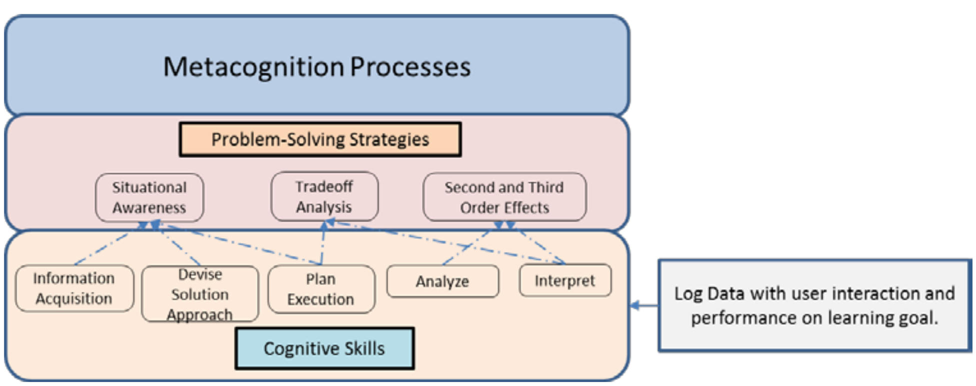


Figure 1. Three-Tier Hierarchical User Model

The middle layer of our user-modeling framework captures users' proficiency on problem solving strategies, which represent meaningful sequences (combinations) of cognitive skills, and provide the link to higher level tasks and goals that the user is trying to accomplish (Kinnebrew, et al., 2016), e.g., clear the area of insurgents to help establish local governance. The top layer captures users' proficiency on metacognitive processes that considers the users' intent in using a skill or strategy, as well as their monitoring, evaluation, and reflection behaviors as they work on the system. Strategies comprise an important component of metacognitive knowledge; they consist of declarative, procedural, and conditional knowledge that describes its purpose and when and how the strategy should be implemented (Schraw et al., 2006). In other words, we conjecture that strategies are an important component of the planning and evaluation phases in metacognition. An important goal that we have adopted in our work is to monitor and infer users' strategies as they work on complex open-ended problems, and provide adaptive support to help them improve their overall performance in their problem solving tasks. Problem solving strategies and metacognitive processes should be defined and apply generally across problem solving tasks. Therefore, we have designed the top layers of the user model to be accessible across different training environments that may be linked to GIFT.

A Hierarchical Task Model

Our OELE task model in GIFT is represented as a directed (acyclic) graph, which provides a successive, hierarchical breakdown of the primary tasks into their component subtasks in the OELE. At the lowest levels of the hierarchy, the tasks are linked to the observable actions in the OELE. The top level of the model identifies the three broad classes of OELE tasks related to: (1) information seeking and acquisition, (2) solution construction and refinement, and (3) solution assessment. Each of these task categories is successively broken down into three levels that represent: (1) general task descriptions that are common across many OELEs; (2) learning environment specific instantiations of these tasks; and (3) observable actions in learning environment through which users can accomplish their tasks.

Implementation of Three-Tier Hierarchical User Model in GIFT

The GIFT environment consists of three primary modules: (1) a user module, (2) a domain module, and (3) a pedagogical module. The domain module contains the domain-specific knowledge file (DKF), which defines (1) the course structure, (2) tasks, (3) concepts, (4) priorities and conditions to assess users' correct application of the concepts, and (5) instructional strategies to provide feedback or adaptive content to the user. In GIFT domain module, the concepts and condition to assess those concepts are represented as flat computational structure. GIFT concepts are assessed using the following performance metrics: (1) *Assessments*, measured as Below, At, and Above Expectation, (2) *Competence*, a value that captures the users' competence on the concept ranging between [0.0, 1.0], (3) *Confidence*, a value that represents the system's confidence in the assessment ranging between [0.0, 1.0], and (4) *Priority*, a unique value that defines the importance of a concept compared to the other concepts, and it is used by the instructional strategy handler to select the concept on which the system provides feedback.

To study how a user's competence evolves as they work on their tasks, we have added a performance metric to GIFT called *Trend* whose values range between [-1, 1]. In our work, we have adopted a simple trend measure that is computed over the user's last two turns.

$$\text{Trend} = Max\left((P_i - P_{i-1}), (P_{i-1} - P_{i-2})\right) \qquad (1)$$

In order to implement, test and validate our proposed user model shown in Figure 1, we modified the GIFT DKF to represent the multiple concept types as a hierarchical structure. The modified GIFT DKF structure is shown in Figure 2. To measure the users' competence on cognitive skills, problem-solving strategies and metacognitive processes, we apply a bottom-up computational approach. First the users' performance on cognitive skills is computed from the parsed data extracted from the log files. The relevant user proficiencies on cognitive skills are then aggregated to derive the competence, confidence, and trend values for problem-solving strategies, which in turn forms the basis for computing these values for metacognitive processes.

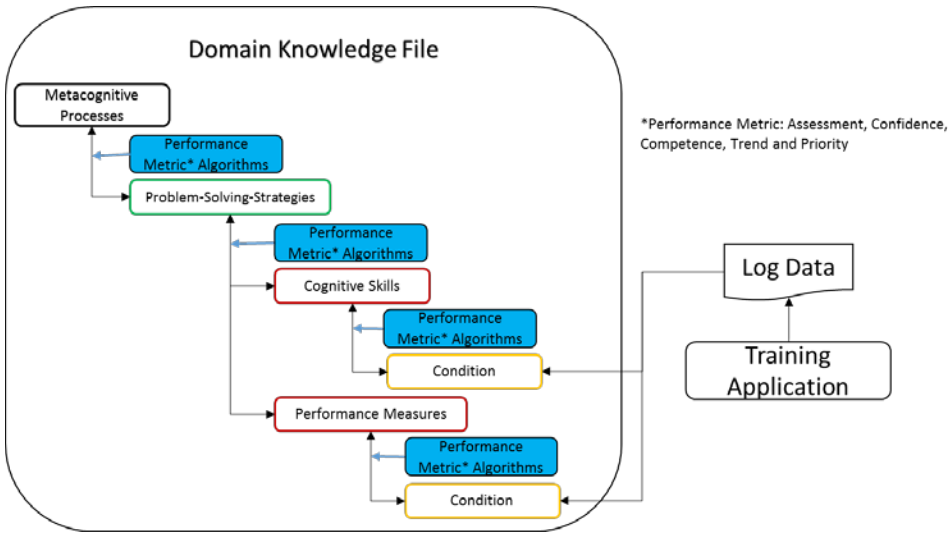


Figure 2. Implementation of Hierarchical User Model in GIFT

Our instructional strategy algorithm builds on this hierarchical structure and the performance metrics that we derive for each level. The algorithm analyzes the performance metrics top-down on the hierarchical structure, starting from the metacognitive level. It checks the competence and trend values of its child nodes (i.e.), the strategy nodes and picks the child node that has the least competence and a negative or flat trend. Then the algorithm repeats the analysis on its child nodes (cognitive skills). If the user is deficient in cognitive skills (i.e., their performance values for cognitive skills are below a threshold), the algorithm will pick one of the cognitive skills for feedback, otherwise, it focuses on the higher-level selected strategy for feedback. If the user has shown sufficient proficiency at the strategy level, then the algorithm selects a metacognitive process node for feedback.

Overall, this hierarchical user modeling structure and corresponding instructional algorithm has significant advantages. First, it explicitly captures the different processes that are important for learning decision-making and problem solving, and the relations between these processes. As a result, problem-solving strategies and cognitive skills are distinguished from domain specific skills, and they can be used across different learning and training systems that are linked to GIFT. More sophisticated instructional strategies can be implemented by combining the bottom-up and top-down analyses, allowing for more accurate assessments of the users, and increasing the scope of the feedback that may be provided to the users. We demonstrate some preliminary work that shows the effectiveness of our approach, but hope to demonstrate the full capabilities of this approach in future work.

IMPLEMENTATION OF THE USER MODEL FOR URBANSIM

In UrbanSim, the overall progress toward meeting COIN goals is represented by an aggregate measure called the Population Support (PS). Population support is derived from the coalition support in the regions for the U.S. armed forces. Users' adherence to the U.S. Armed forces Brigade Commander's intent or goals for the counterinsurgency operations, are measured as six Lines of Effort (LOE) scores. The LOE values are an aggregate of the Political, Military, Economic, Social, Information, and Infrastructure (PMESII) scores associated with each region in the area of operations (Tscholl, et al., 2016 & Tscholl, et al., 2016b).

As discussed earlier, user behaviors in the UrbanSim environment are derived from their interactions with the environment captured in the form of log files. To interpret users' activities in UrbanSim within the GIFT tutoring framework, we classify their interactions with the system into two types: (1) *Actions* and (2) *Operations*. Actions represent users' interaction with UrbanSim recorded as messages in the UrbanSim log files. UrbanSim permits 43 different actions related to acquiring information in the COIN scenario. Example actions include view Intel reports and study trends (graphs). Operations are commands assigned to the units to perform on a region or individual. UrbanSim has repertoire of 21 different counterinsurgency operations that apply to an area of operation (AO). Example operations include patrol neighborhood, cordon and search, and host meetings with specific individuals.

The Hierarchical Task Model in UrbanSim

Figure 3 illustrates the hierarchical task model we have created for the UrbanSim counterinsurgency game. At the lowest level, subtasks related to information acquisition, solution construction, and solution assessment are linked to users' observable behaviors (actions and operations) in UrbanSim. Information seeking and acquisition involves understanding mission goals, identifying, evaluating the relevance of, and interpreting information to make logical decisions in the context of the overall mission goal and the current task. Solution construction and refinement tasks involve applying information gained to devise the solution approach (e.g., changing the actions in the sync matrix). Finally, solution assessment tasks involve interpreting the results of metrics provided by the system, such as the PS, LOE, PMESII, and CS values, to analyze, and if necessary, refine the solution approach.

The task model structure with accompanying computational algorithms are used to infer the users' generic cognitive skills. Similarly, we measure the users' proficiency on domain specific cognitive skills, i.e., their understating of clear (C), hold (H) and build (B) based on the operations selected by the user for their last few turns in the game. In order to measure the C, H, and B values for each turn, the regions in the AO are classified as clear, hold or build based on their corresponding Political and Military values. We adopted the classification of clear, hold, and build operations from (Vogt, 2012, p. 50) and also by consulting our domain experts in the ROTC program at Vanderbilt University. At every turn, we check to see if the clear, hold, and build operations are conducted in appropriate regions, and use this analysis to compute the users' proficiency in C, H and B.

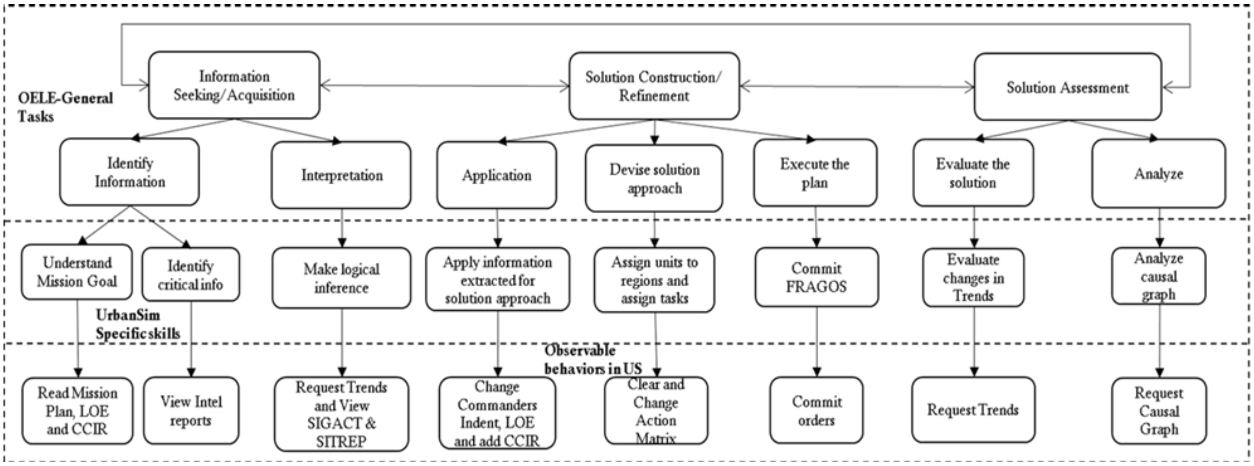


Figure 3. UrbanSim Hierarchical Task Model

Problem-Solving Strategies in UrbanSim

The goal of the UrbanSim learning environment is to help users understand the complexity of Counterinsurgency (COIN) operations, and apply strategies and decision-making skills to overcome these complexities. Successful application of the CHB strategy in complex counterinsurgency environments requires the trainee officer to develop and apply a number of additional strategies. In our work, we have identified three such strategies.

1) ***Situational awareness (SA)***: ability to identify and interpret key information in the Area of operations (AO) and develop a common operating picture (COP). This requires performing mission analysis as described in (McAlinden, et al., 2008).
2) ***Trade-off analysis (TA)***: a methodology for choosing operations taking into account the limited resources available, such as CERP funds, and units to conduct operations. Typically trade-off analysis may involve prioritizing operations to be conducted in different regions of the AO and then selecting and performing the high priority operations, while setting aside lower priority ones**.**
3) ***Second and third order effects (SOE)***: analyzing and predicting the effects of operations that are compatible with a prescribed end goal. An important consideration here is the decision to conduct lethal versus non-lethal actions realizing the direct and indirect effects that each of these operations may have on future CHB operations. Analyzing second and third order effects contributes significantly to both mission planning and evaluation.

In order to solve a complex task, it is clear that these strategies cannot be applied independently. For example, the user may first apply SA to generate a COP that takes into account the different regions, their military, political, and economic status, as well as the actors of significance in the current COP. A more complete understanding of the COP helps the user select appropriate COIN operations using the CHB strategy. However, given the limited number of personnel and budget available, the user has to determine a number of relevant operations that could be enacted, but perform tradeoff analysis between possible courses of actions to pick ones that best match the commander's intent (increase LOE values) and contribute to the overall goal, i.e., PS. The effects of the operations selected are provided as performance feedback and trend analysis to the user. In addition, after a few turns they may also review the effects of past operations to study second and third order effects, which can govern the selection of future operations.

GIFT User Model for UrbanSim

In order to analyze the users' proficiency on COIN operations in UrbanSim, we first analyze the Brigade commander's intent provided to the user at the start of the game. For example, this analysis provided the key goals to pursue in the Alhamra-2 scenario (a fictional Iraqi city, based very loosely on the conditions seen in the northern Iraqi city of Tal Afar): (1) stop influx of insurgents into the area of operations (AO), (2) improve effectiveness of utilities, such as, water, electric, trash depot and sewage plant, (3) ensure that the Iraqi security force (ISF) have local, reliable leadership, and is adequately funded and trained, and (4) build the regions' infrastructure to achieve economic viability. The performance on these key tasks are measured using performance measures Users' performance on these key metrics are extracted from UrbanSim log file.

We implemented GIFT user model for UrbanSim to measure the user's performance on cognitive skills, problem solving strategies, and metacognitive processes for each turn. For lack of space, we do not discuss the assessment of all problem-solving strategies in this paper. Instead, we discuss the algorithms for measuring users' Situation Awareness and Tradeoff Analysis strategies.

Tradeoff Analysis and Situational Awareness

Tradeoff Analysis measures the users' decision-making proficiency in choosing current operations, taking into account the limited amount of funds and personnel that are available to them. Tradeoff analysis focuses on the user's ability to choose operations that create a balance between performance measures, such as Effectiveness, Military Power, Capacity, and Coalition Support. Therefore, given the limited resources, trade-off analysis may involve prioritizing operations to be conducted in regions, and then performing the operations that maintain priority and balance. We have developed an algorithm (Algorithm 1) that uses information from the UrbanSim log data to compute the user's competence in TA. The first step of the algorithm classifies the AO regions as clear, hold or build using their corresponding Political and Military values. Overall the AO is categorized to be in a particular phase (C, H, or B) if the number of regions in that category exceed that of the other two. The TA measure for a user is then incremented if the actions and operations performed lead to an increase of the corresponding metrics as shown in Algorithm 1.

Algorithm 1: To measure User's Proficiency on *Tradeoff Analysis*
Input: Turn number*, MP of Insurgent and Iraqi force, CP of Infrastructure, EF of utilities, CS, and PMESII*
Output: User Proficiency on *TA*

For each Turn
1. For number of regions**, do**
  Classify the region as Clear , Hold or Build Phase by its P and M values
 end
2. If Number of Regions in Hold Phase > Regions in Clear Phase and Regions in Build Phase
  if Delta of CS > 0 and Delta of EF > 0
   Increment TA competence by 0.1
  else Decrement TA competence by 0.1
 If Number of Regions in Build Phase > Regions in Clear Phase and Regions in Hold Phase
  if Delta of CP > 0 and Delta MP(Police) > 0 and Delta of CS > 0
   Increment TA competence by 0.1
  else Decrement TA competence by 0.1
 end

Algorithm 1: To measure users' competence on Tradeoff Analysis in UrbanSim environment.

In general, if a user only selects operations that improve one or the other, then their ability to perform tradeoff analysis must be low. For example, if user performs operations that improve the values of capacity of infrastructure, military power of Iraqi security force and coalition support, then the user demonstrates good TA strategies.

Situational awareness (SA) measures the users' ability to identify and interpret key information in the area of operation (AO). In order to measure users' competence in SA, we analyze their performance in meeting the brigade commander's intent using performance measures, for example, the Coalition Support (CS), and Effectiveness (EF) of utilities. A sample of our algorithm to measure users' proficiency in SA during Hold phase is given in Algorithm 2. If we detect that the trend of any performance measure is negative, we consider the user's SA performance is low, hence, we further analyze the users' awareness of the environment in which they are operating. To understand the environment, the user needs to perform actions, such as read the mission statement, and understand the political, economic and military network of individuals and groups in the AO. We operationalize these actions as subtasks as described in the task model, Figure 3. List of subtasks required for SA are Identify, Interpret, Apply and Devise Solution. We check, whether the user has done these subtasks and then performed operations to improve the performance. If the user has performed these subtasks and trend of any performance measure is negative then we trigger a conversation with the user to further probe their awareness. Based on users' response in the conversation tree the competence of SA is updated.

```
Algorithm 2: To measure user's proficiency on Situational Awareness
Input: MP of Insurgent and Iraqi force, CP of Infrastructure, EF of utilities, CS, and PMESII
Output: User proficiency on SA values
Initial SA competence = 0.0
For each Turn
1.    For  number of regions, do
            Classify the region as Clear , Hold or Build Phase by its P and M values
      end
2. If Number of Regions in Hold Phase > Regions in Clear Phase and Regions in Build Phase
       If Delta of CS > 0 and Delta of EF > 0
                    Increment SA competence by 0.1
       else
       If  Average of Tasks(Identify, Interpret, Apply and Devise Solution) > 0.3
                   Start Conversation to user to check coherence between their action and operation.
                   Update SA competence based on user's response.
       else
end
```

Algorithm 2: Snippet of algorithm to measure users' competence on Situational Awareness

RESULTS

We validated the user model by correlating users' proficiency on problem solving strategies with their performance in domain-specific task.

Participants: Fourteen senior ROTC students at Vanderbilt University participated in our study. These students worked in pairs during two separate 2-hour sessions (approximately one month apart). Due to absences, these 14 students made up eight different groups over the two sessions, with four groups remaining the same for both sessions.

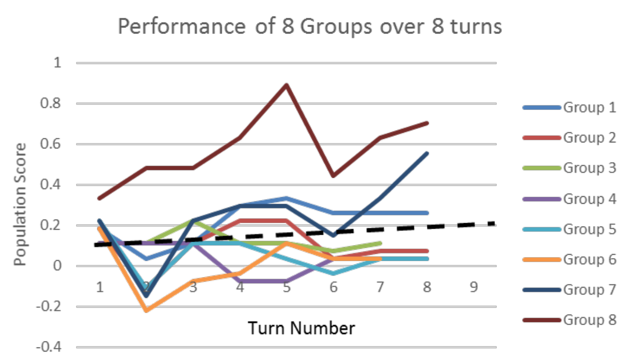


Figure 4: User performance on Population support for 8 turns

Procedure: Students used UrbanSim to practice and apply their knowledge of COIN principles in two scenarios: Al-Hamra and Al-Hamra 2. The study contains following steps (1) pre-test to assess user's under-standing of COIN principles and operations, (2) Students worked on the Al-Hamra 2 scenario for approximately 90 minutes, (3) After a break of about four weeks, students worked on the Al-Hamra scenario for approximately 90 minutes followed by (4) post-test. In both sessions, the course instructor led a debriefing discussion with the students after they had worked on the two scenarios. Students' interaction with learning environment and their discussions during the study are recorded using the Camtasia software.

Analysis: First, we analyzed the performance of all groups on population support for 8 turns as shown in Figure 4. Based on their performance, we separated the groups into high and low performers, as indicated by the dotted line in Figure 4. Then we selected one group from each category for detailed analysis. Figure 5, shows, for two groups, the high performing (group 7) and the low performing (group 2), their proficiency on strategies, population score and how they evolved as they interacted with the system for 13 turns. In Figure 5, the users' proficiency on SA of both groups is computed. At this stage they look to be about the same, however, group 2 was not able to maintain this proficiency level on their TA.

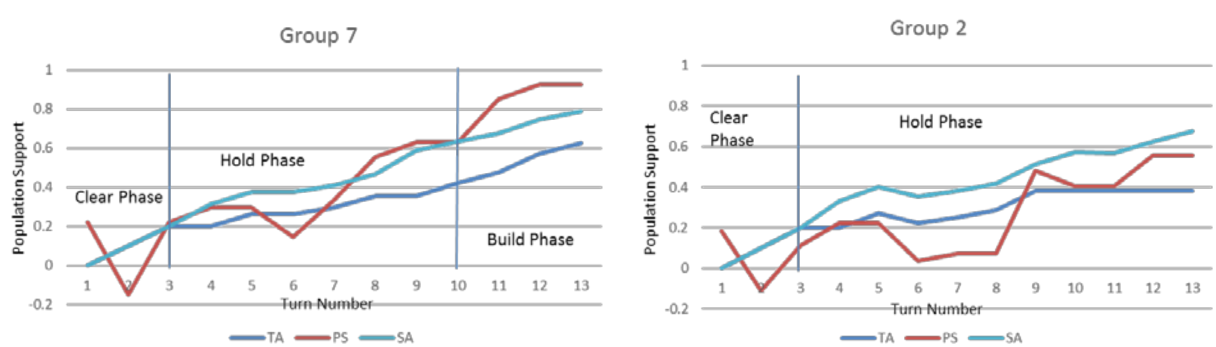


Figure 5. Two User Groups Proficiency on Tradeoff Analysis, Situational Awareness and Population Support

In the Clear phase, both groups chose operations to clear regions, and balanced use of resources. Therefore, their TA proficiency is increasing for every turn. In the Hold phase, group 7, chose operations that balanced Effectiveness of utilities versus Coalition support, hence their performance on population support (PS). Their choice of balanced operations is indicated by improved TA performance in the Hold and Build phases. As a result, they made smooth transitions from the Clear to the Hold to the Build phase. On the other hand, Group 2 did not choose operations that balanced Effectiveness versus Coalition support in the hold phase, hence their performance deteriorated and they could not make the transition from the Hold to the Build phase.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

In this research work, we proposed and implemented a user model using GIFT to measure the users' proficiency on cognitive skills and problem solving strategies, which can be used across different learning environments. Moreover, from our analysis we developed an instructional strategy algorithm to provide feedback to users based on their proficiency on cognitive skills and problem-solving strategies. In future, we propose to develop algorithms to measure the users' proficiency on metacognitive processes by analyzing the proficiency and trend values of cognitive skills, problem solving strategies and their performance in the domain-specific task and conduct research study to validate our proposed user model. Inferring metacognitive processes is a challenging task, since it happens in the users mind. Therefore, we are developing a combination of detection and querying methods to infer users' metacognitive processes online as they work on the system.

REFERENCES

Hannafin, M. J., Hall, C., Land, S. M., & Hill, J. R. (1994). Learning in open-ended environments: Assumptions, methods, and implications. Educational Technology, 34, 48-55.

Kinnebrew, J., Segedy, J., & Biswas, G. (2016). Integrating model-driven and data-driven techniques for analyzing learning behaviors in open-ended learning environments, IEEE Transactions on Learning Technologies (In press). DOI: 10.1109/TLT.2015.2513387

Kulik, J. A., & Fletcher, J. D. (2016). Effectiveness of Intelligent Tutoring Systems A Meta-Analytic Review. Review of Educational Research, 86(1), 42-78.

Lester, J., Lobene, E., Mott, B., & Rowe, J. (2014). –Serious Games with GIFT: Instructional Strategies, Game Design, and Natural Language in the Generalized Intelligent Framework for Tutoring. Design Recommendations for Intelligent Tutoring Systems, 2, 205-215.

McAlinden, R., Durlach, P., Lane, H., Gordon, A., & Hart, J. (2008). UrbanSim: A game-based instructional package for conducting counterinsurgency operations. In: Proceedings of the 26th Army Science Conference, Orlando, FL.

Nagl, J. A., Amos, J. F., Sewall, S., & Petraeus, D. H. (2008). The US Army/Marine Corps Counterinsurgency Field Manual. University of Chicago Press.

Schraw, G., Crippen, K. J., & Hartley, K. (2006). Promoting self-regulation in science education: Metacognition as part of a broader perspective on learning. Research in science education, 36(1-2), 111-139.

Segedy, J. R., Kinnebrew, J. S., & Biswas, G. (2015a). Coherence over time: understanding day-to-day changes in students' open-ended problem solving behaviors. In International Conference on Artificial Intelligence in Education (pp. 449-458).

Sottilare, R. A., Brawner, K. W., Goldberg, B. S., & Holden, H. K. (2012). The generalized intelligent framework for tutoring (GIFT). https://gifttutoring.org/projects/gift/wiki/Overview

Tscholl, M., Biswas, G., Goldberg, B. S., & Sottilare, R. A. (2016). Automated Detection of Cognitive and Metacognitive Strategies for Learner Modeling in GIFT. In Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym4).

Tscholl, M., Biswas, G., Goldberg, B. S., & Sottilare, R. A (2016b). Detecting Metacognitive Strategies through Performance Analyses in Open-Ended Learning Environment. In Chen, W. et al. (Ed.), Proceedings of the 24th International Conference on Computers in Education.

Vogt, B. D. (2012). A Methodology to Assess UrbanSim Scenarios, Master Thesis, NAVAL POSTGRADUATE SCHOOL MONTEREY CA.

ABOUT THE AUTHORS

Dr. Ramkumar Rajendran *is a Postdoctoral Researcher at the Institute of Software Integrated Systems at Vanderbilt University.*

Naveeduddin Mohammed *is a Research Engineer at the Institute of Software Integrated Systems at Vanderbilt University.*

Dr. Gautam Biswas *is a Professor of Computer Science, Computer Engineering, and Engineering Management in the EECS Department and a Senior Research Scientist at the Institute for Software Integrated Systems (ISIS) at Vanderbilt University.*

Dr. Benjamin Goldberg *is an adaptive training scientist at the Army Research Laboratory's SFC Paul Ray Smith Simulation & Training Technology Center. He leads research focused on instructional management within ARL's Learning in Intelligent Tutoring Environments (LITE) Lab and is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT).*

Dr. Robert Sottilare *leads adaptive training research within ARL's Learning in Intelligent Tutoring Environments (LITE) Lab and is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT).*

Physiological Based Adaptive Training

Thomas Schnell[1], Cory J. Reuter[1], Ezekiel D. Gunnink[1], Bradley M. Parker[1], Carlton H. Richey[1], Jaclyn A. Hoke[2], Jason D. Moss[3]
Operator Performance Laboratory (OPL)[1], Rockwell Collins[2], US Army Research Laboratory (ARL)[3]

INTRODUCTION

The Army faces an emerging adversary environment that is very competitive, dangerous, and cognitively intense. To address this challenge, Army soldiers must out-learn and out-train their adversaries and this training challenge must be met in a climate of austere or shrinking training budgets (Army, 2011a, 2011b). The gold standard in training is one-to-one human tutoring, which has been shown to be significantly more effective than the one-many method of instruction such as the traditional classroom setting or self-study using static training materials such as manuals and books (VanLehn K, 2011). The proliferation of computer based games including massively multiplayer online games (MMOG), low-cost simulations, and exciting virtual immersion technologies opens new doors in the training domain. Additionally, considerable progress has been made in areas that include training pedagogy, methods of instruction/feedback, artificial intelligence, virtual humans, and trainee state assessment. Through a well-crafted learning concept roadmap, the Army plans to leverage those technological game changers to create systems that will allow self-paced, adaptive training capabilities that will enhance training effectiveness while at the same time be very cost effective. To address this challenge, the Army Research Lab has developed the Generalized Intelligent Framework for Tutoring, which is known as GIFT (Sottilare R, Sinatra A, & Boyce M, 2015). In our project, we are adding a component to GIFT that uses the Cognitive Assessment Tool Set (CATS) as a system to acquire the real-time operator state (Ellis K.E, 2014; T. Schnell, 2012; T. Schnell & Engler, 2013; T. Schnell, Melzer, & Robbins, 2009). This includes task technical performance, cognition (workload, engagement), and attention (degree of focus), so that the training content can be adapted through GIFT to maximize training effectiveness. We selected a demonstration use case centered on self-study driving instruction training for military vehicles, particularly for the High Mobility Multipurpose Wheeled Vehicle (HMMWV). Initially, we are using the Virtual Battle Space (VBS3) as the driving simulation tool (Figure 1a). In Year 2 of this project, we will migrate the GIFT Framework into our instrumented Model 997 HMMWV Off-Road Testbed for testing in a real-world off-road environment (Figure 1b).

a. VBS 3 HMMWV Simulator

b. OPL's Instrumented HMMWV/Simulator

Figure 1. Physiological Based Adaptive Training using GIFT Framework for HMMWV

The training scenarios in our adaptive training concept progress in difficulty from simple driving tasks on a flat and level tarmac to complex urban navigation and off road maneuvering assignments. This specific use case was developed after various training domains and application domains have been reviewed, alternatives were developed, and a down-selection was performed to arrive at the particular driver training use case which will form the basis of the testbed in this project.

THE PROBLEM

Current training tools do not usually have an ability to acquire trainee data beyond simple performance data (e.g. right and wrong answers). Therefore, current training systems are generally not able to associate trainee state to specific elements of instruction. In one-to-one human tutoring settings, the instructor observes the student's performance and exterior performance indicators such as body language, facial expressions, head position, and hand movements to make a determination if the trainee is on the right path to acquiring the skill. For example, an Instructor Pilot (IP) may carefully and unobtrusively observe his/her flight students during landings to see if they are referencing the correct instruments and perform the correct manual movements for this phase of flight. Through such exterior performance observations, the IP can assess trainee state in real time and take corrective action, if necessary. Such actions could include physical interventions (e.g. to prevent a crash), the provision of explanations, a decision to repeat the task, or a decision to abandon the task and allow the student to rest. Unfortunately, there are circumstances where it is impossible for an instructor to discriminate with external indicators alone, if a student has frozen up or if he/she is cool and in control but is not currently moving around. Additionally, instructors are a limited resource and it is not feasible to have one-on-one tutoring in all training settings.

Therefore, the Army is looking for a data driven approach that will automatically and unobtrusively assimilate trainee information and then reliably and automatically classify trainee state including performance, cognition (workload, engagement), attention (degree of focus), and affect (joy, confusion, frustration, boredom, surprise, and anger) so that the training content can be adapted to maximize training effectiveness.

PHYSIOLOGICAL BASED TRAINEE STATE MODULE

In the project described in this paper, we are adding a component to GIFT that uses the Cognitive Assessment Tool Set (CATS) (OPL, 2014) as a system to acquire the real-time cognitive workload of the trainee to close the loop, through GIFT, with the training application. This means that the workload experienced by the trainee affects the progression of the training application. We selected a demonstration use case centered on self-study driving instruction training for military vehicles, particularly for the HMMWV. We are using the Virtual Battle Space (VBS3) as the driving simulation tool. We call this combination of GIFT, CATS, and a training application, in the specific use case VBS3, the Unobtrusive Physiological Classification and Adaptive Training (UPCAT) system. This specific use case was developed after various training domains and application domains have been reviewed, alternatives were developed, and a down-selection was performed to arrive at the particular use case which will form the basis of the testbed in this project. In Year 1 of this project (current year), we are integrating CATS and GIFT with the Virtual Battle Space (VBS 3) simulation tool. This constellation will allow us to test the adaptive capabilities of GIFT instruction on the basis of a simulated driving task. In the following project year (Year 2), we will migrate the framework into our instrumented Model 997 HMMWV (see Figure 1). This vehicle can be used as an Automobile-In-Loop (AIL) simulator and it can also be driven on and off-road as a human factors driving research testbed.

Cognitive Assessment Tool Set (CATS)

Understanding and monitoring the changes in the cognitive workload of trainees can offer critical quantitative information about their progression and performance. Unfortunately, accurate real-time objective quantification of cognitive workload using physiological signals has, thus far, proven elusive and is often neglected in favor of subjective self-reports. In well over a decade of physiological based assessment work, we investigated many sensors and came away with the conclusion that the electrocardiogram (ECG) waveform is by far the best signal for workload assessment. Based on our extensive real-world data collection experience, we discourage the use of invasive sensors such as electroencephalogram (EEG) for operational training contexts. The test-retest validity of these EEG appliances is usually very poor, approaching chance probability of prediction. For our ECG based workload assessment, we are using a deterministically nonlinear dynamical classifier to assess cognitive workload with great success (Engler & Schnell, 2013; T Schnell & Engler, 2014). The research community has known for a number of years that human physiological signals in general, and ECG specifically, are deterministically nonlinear (also known as chaotic) systems (Govindan, Narayanan, & Gopinathan, 1998; Kozma, 2002; Owis, Abou-Zied, Youssef, & Kadah, 2002). Chaotic systems are often not well represented via the normal scalar time series. Instead, the dynamics of the system are obfuscated in the single dimension whereas they become apparent when a transform of the data is made. This transform moves the data from the single dimensional scalar space into a multi-dimensional embedded phase space (Richter & Schreiber, 1998). The transformation to phase space using the mutual information and false nearest neighbor techniques can be illustrated nicely with an ECG signal. The panel on the left of Figure 2 depicts a portion of an ECG signal from a subject in a recent study. After calculating the parameters as described above, the phase space can be generated with time delay $\tau = 8$ and embedding dimension $d = 3$. The panel in the middle of Figure 2 shows the phase space that is generated from the signal using the methods described above. The image of the phase space does not necessarily illicit new knowledge about the ECG signal in and of itself. However, the phase space can be coarse-grained (right panel in Figure 2) into a numerical array that represents a quantitative signature of operator state in ECG phase space and thus offers the possibility for accurate operator state characterization. We refer to this as the Chaotic Physiological Classifier (CPC) method.

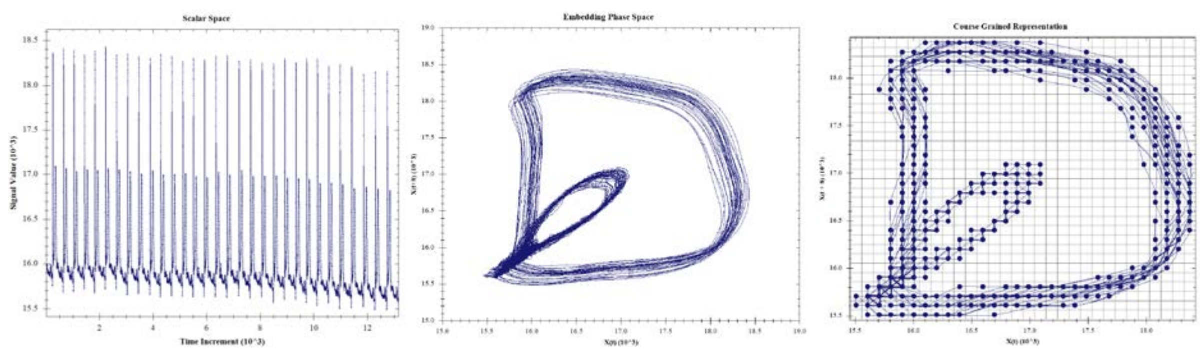


Figure 2. Example of a Scalar ECG (left) Transformed into Embedding Phase Space (middle) and then Coarse-Grained for Numerical Classification (right).

The CATS backbone is a relational database which forms the repository of all data collected during a study. Data is collected via providers within CATS which form communications links between the relational database and the sensing hardware. All data that is collected in CATS is time stamped at the source of the data with a globally synchronized time stamp. The relational database is then structured such that the time stamps form a candidate key for each table, thereby inherently synchronizing the data as it is recorded. Multiple tables exist within the relational database representing multiple signal sources such as vehicle state, environmental (simulator) state, eye tracking, and ECG. Each of these tables is linked through foreign keys indicating which subject, vehicle, and task to which each record is linked. This form of candidate and foreign keys forms a robust, indexable data backbone for the operator state classification effort. In addition to recording data, the CATS system calculates certain metrics in real-time. These metrics are then available to be shared, in real-time, with research partners through the previously mentioned communications portals. CATS uses a CPC model to produce the workload classification based upon the real-time input of ECG data.

UPCAT System Architecture

Figure 3 shows the architecture of UPCAT and its connectivity to the GIFT framework. As shown in this (greatly simplified) diagram, CATS receives data from the ECG sensor (Nexus 4 made by MindMedia) through its standard sensor provider that makes it manufacturer independent. Inside of CATS, the Chaotic Physiological Classification (CPC) (OPL, 2014; T Schnell & Engler, 2014) embeds the time-series ECG data in phase space and applies the ergodicity classification to it. In this context, it is easiest to think of CATS as a processor that translates full ECG waveforms to cognitive workload numbers. The real-time workload number is passed to a processor in CATS which aggregates the rapidly fluctuating number into a relatively stable score that indicates the degree of trainee engagement in the task. This score is transmitted to the Workload Condition in GIFT. The workload measured indicates a type of effort expenditure of the trainee. This expenditure yields a level of driving performance as a function of experience level. A novice driver may expend a significant effort to achieve a relatively low level of driving performance. As training iterations are performed at a certain difficulty level (as driven by the scenario), the effort expenditure should decrease and the driving performance should increase. At some point, both metrics may plateau and if driving performance and workload expenditures are considered satisfactory, the trainee is advanced to the next level of difficulty, either within the scenario or by switching to a more difficult scenario in VBS3. Driving performance is assessed through quantitative metrics that relate to automatically measurable outcomes such as speed maintenance, lane control, steering wheel rotation entropy, etc.

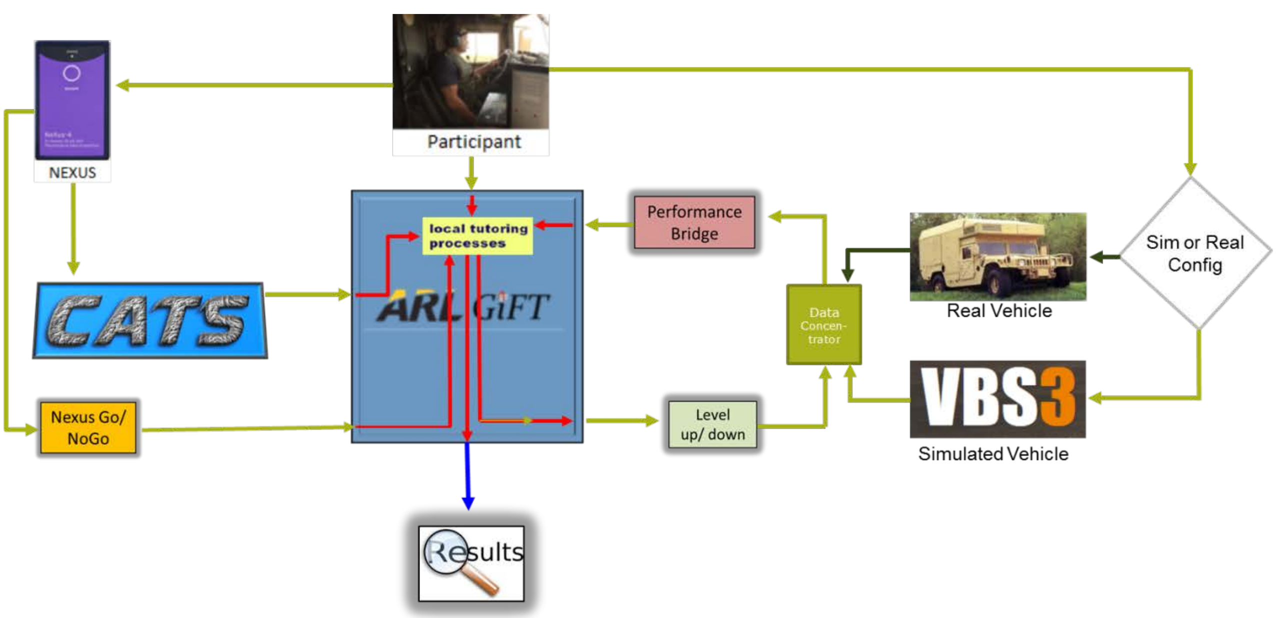


Figure 3. UPCAT System Architecture

The VBS 3 training scenarios in UPCAT progress in increasing levels of difficulty, much like advancing through chapters of a book. At strategic points in the scenarios, the driving events are stopped (frozen) and the trainees fill out surveys provided by GIFT to indicate the level of self-rated performance and workload expenditure. After the surveys are completed, the responses are combined with the physiological based data and the driving performance results to make decisions on how the scenario should proceed. In GIFT, the following steps are needed to accomplish this: 1). Create the workload scoring filter, 2). Create the workload scoring condition, 3). Create the VBS3 scoring filter, 4). Create the VBS3 scoring condition, 5). Create the surveys per scenario, 6). Create the survey scoring condition, 7). Create the real time difficulty changing interface (VBS 3 scenarios), 8). Combine the workload and VBS3 filters into a Domain Module, 9). Transmit the information via a Learner Module into a Pedagogical Module so that we can act on these conditions as a group, 10). Create a Domain Knowledge File (DKF, a set of rules for performance) and author Triggers to change scenarios and trigger difficulty changes. GIFT has a VBS3 interface that allows transmission of commands to a running VBS3 instance. These commands include calling a script remotely and reading out the results of its execution. In our architecture, this facility is used to extract driving performance score calculations out of VBS3.

The GIFT system is implemented as constellation of state machines in different states. The GIFT Intermission Stage (shown in Figure 4a) handles the transitions between scenarios by analyzing the results of previous scenarios. This allows us to adaptively increase or decrease the difficulty of the scenarios as a whole unit. The GIFT Run Stage in (shown in Figure 4b) adjusts the difficulty within the current scenario in real time as it is being run. These adaptive changes in difficulty will be smaller incremental changes than when the scenario difficulty is changed as a whole unit. Scripts inside of VBS 3 respond to changes in scenarios and difficulty according to the script state machine (shown in Figure 4c) in response to the other GIFT stages.

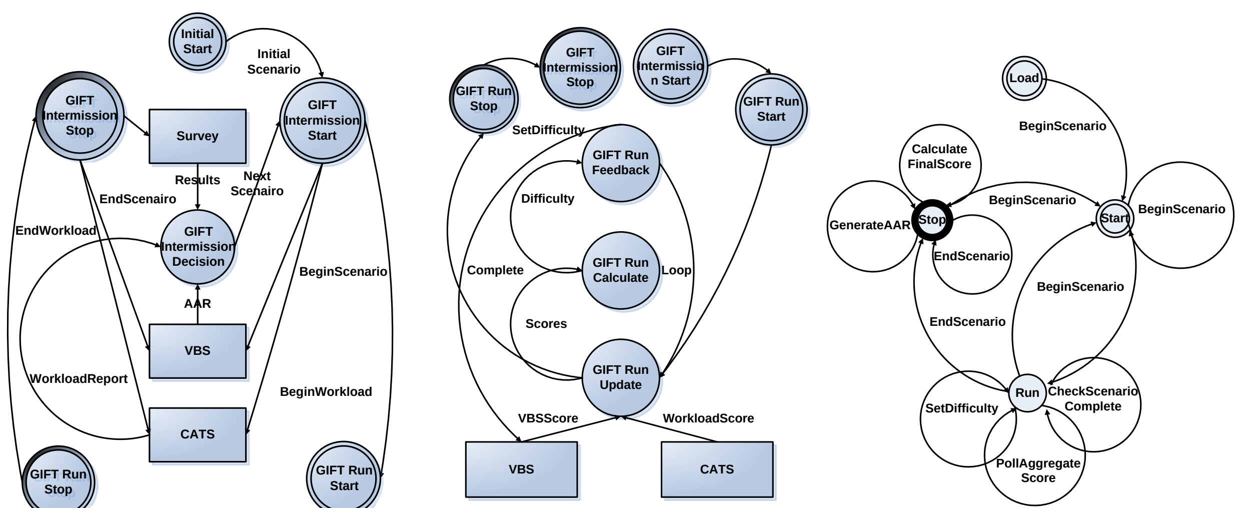


Figure 4. State Machine Diagrams for GIFT Implementation in UPCAT

UPCAT CONCEPT OF OPERATIONS (CONOPS)

The following is the CONOPS narrative that describes what the UPCAT system, once completed and when applied for the HMMWV driving use case, should be able to do. This CONOPS has driven our architecture design and will help us to complete the UPCAT system in accordance with set requirements. The CONOPS is a narrative that describes how we want the finished UPCAT system to work. While the

CONOPS is relatively detailed and thorough, we will focus the effort in this project on the physiological based adaptive training capability and only “rough in” some of the training environment capabilities described in this CONOPS. We developed a detailed visual storyboard that describes the graphical content of the UPCAT screens that the trainee would see. Some selected images from that storyboard are represented as figures hereinafter.

The expected trainee is an army recruit who has a valid US driver license and about 3-4 years of on-road driving experience on normal US highways under 4 season day and night driving conditions. As a baseline, we assume that the trainee has no prior off-road driving experience and no driving experience in foreign countries.

The trainee is assigned to an UPCAT workstation where he/she logs in and starts the enrollment process using an interactive screen to fill in information. The trainee enters pertinent information about his/her person to facilitate tracking of course credit. Additionally, the trainee enters information related to his/her driving experience such as number of years driven, area where driving was performed, urban vs rural driving, day vs nighttime, driving on snow, type of vehicle, etc. This is done to establish a baseline database of driving exposure. The UPCAT system then provides the trainee with an instructional video that illustrates how the UPCAT ECG sensor is to be applied. This shows attachment of the electrodes using a schematic view of a person’s torso to be sure the electrodes and leads are attached correctly. The video then stops to give the trainee a chance to set up the electrodes and go to the next screen. UPCAT then tells the trainee how to start CATS and verify ECG signal accuracy (Figure 5). Next, the trainee goes through a set of slides that introduce the HMMWV. This is basically a Computer Based Training (CBT) user manual review introducing the HMMWV controls. Once the trainee completes the basic CBT, a quiz will be administered (Figure 6) to ensure the trainee is ready to progress to the first driving simulator module.

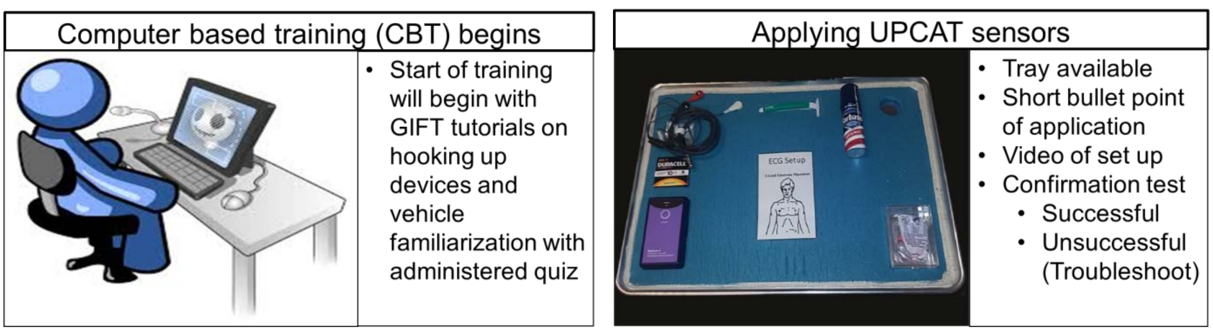


Figure 5. UPCAT CBT and Sensor Application

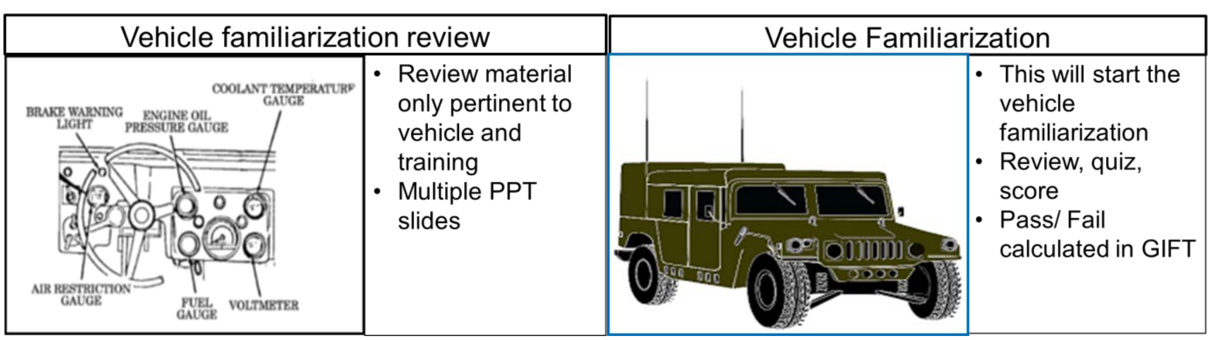


Figure 6. UPCAT Vehicle Familiarization

The first driving simulator module (Level 1) is a simple drive on a very large parking lot or tarmac without obstacles. Using graphical interactive content (Figure 7, left) the trainee is told to drive around the parking lot perimeter in a clockwise direction, one car width away from the apron edge, at an appropriate speed, not to exceed 20 MPH. Performance are measured by UPCAT to ensure that the trainee has maintained the speed and positional assignment. CATS is used to assess workload to ensure that sufficient replications of the drive around the tarmac have been completed. The trainee is considered ready for the next level when workload has levelled off and driving-technical performance is within boundaries. A point score is then calculated from the performance metrics as credit similar to the score in a video game (Figure 7, right).

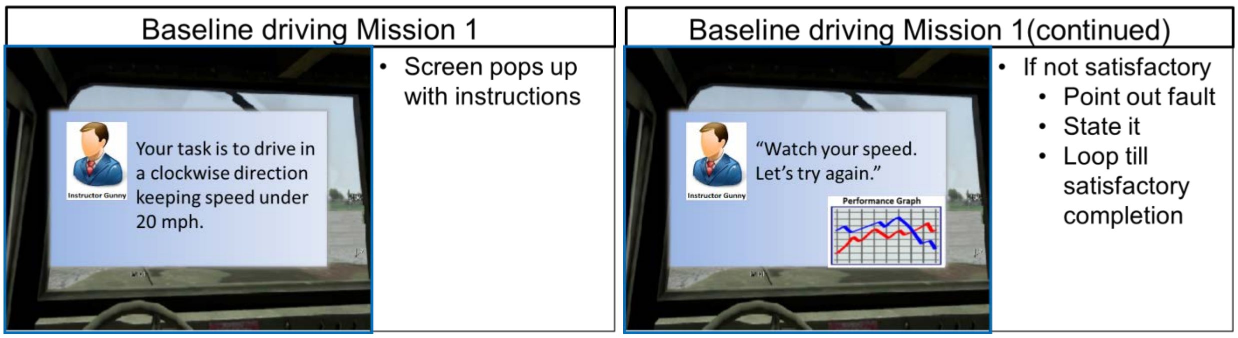


Figure 7. UPCAT Simple Driving Task (Level 1)

Upon completion of the first level, the point score and other statistics are shown. Feedback is provided by GIFT using trigger points such as a) great job, all is well, b) watch your speed, c) watch your lane control etc. These feedback points are illustrated with performance graphs and verbal narratives from canned AVIs playing an instructor (Gunny) chastising or praising the student (Figure 7, right). The trainee then proceeds to the next level and the process repeats for as many levels as needed by the particular use case. In our project, the progression of scenarios may look as follows:

Level 2: Parking lot with obstacles placed to drive around.
Level 3: Driving on an open, mostly straight highway in a foreign country with appropriate visuals and a simple navigational assignment.
Level 4: Addition of curves and reasonable up and down grades.
Level 5: Addition of urban areas.
Level 6: Addition of roadway threats to avoid, requiring severe braking and swerving.
Level 7: Addition of off-road, straight up and down grade.
Level 8: Addition of off-road, along grade (slant), left and right.
Level 9: Addition of driving at nighttime and in degraded visibility conditions.
Level 10: Addition of IED detection and avoidance.
Level 11: Addition of ambush event with backup retreat.
Level 12: Capstone driving event that is assembled from all the parts that the trainee did not do well on.

Provisions should be made so that a driving session can be interrupted and taken up again. We still will need to determine how feasible this is with regard to physiological based workload assessment without baseline. It may be necessary to repeat the level upon resuming after a long break (e.g. days).

Once all levels on the driving simulator have been completed, the GIFT training record is forwarded to the driving instructor for review and scheduling of the first live driving lesson. The idea of the live drive is that UPCAT rides along, permitting the use of a safety observer who is not a qualified driving instructor but rated in the vehicle only. Therefore, staffing is easier because the safety observer does not need to have the qualities of an instructor as that job will be done by UPCAT. Instead, the safety observer simply monitors the drive with regard to safety. In the live drive, UPCAT receives vehicle state not from VBS3 but from the vehicle inertial system.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

The GIFT architecture facilitates the integration of external tools such as VBS 3 and CATS in a very effective fashion. CATS is an operational workload and performance assessment system that has been used by OPL in real-world driving and flight contexts for a number of years. CATS has been used many times to assess the performance of fighter pilots in OPL’s instrumented jet aircraft or in flight simulators at OPL and numerous government research facilities. In this project, we are using the workload assessment capability of CATS and integrate it with the GIFT framework using a Direct Link Library (DLL) methodology.

At the time of writing this paper we are about 5 months into the first program year. We have finished the architecture design and implemented an initial prototype in accordance with Figure 3. In the remainder of Year 1, we will complete the initial UPCAT prototype and demonstrate the physiological based adaptive training scenario capability.

For the Year 2 effort, we are planning to test and evaluate the UPCAT system using N=12 participants undergoing a full-mission training evolution as described in the CONOPS. There are several research questions that this experiment will seek to answer.

1. Is the UPCAT workload assessment accurate (absolute) and precise (narrow distribution) when compared to a known or self-assessed baseline workload scale?
2. Is the UPCAT affect assessment accurate (absolute) and precise (narrow distribution) when compared to a self-assessed baseline affect scale (e.g. joy, confusion, frustration, boredom, surprise, and anger)?
3. Is the adaptive portion of the training more effective than its non-adaptive counterpart?
4. Is the base program UPCATS/GIFT system acceptable for actual training in an Army context?

Research questions 1-3 will be answered though experimental hypotheses resulting from a full factorial experiment with assessments performed using appropriate statistical tests. Research question 4 will be answered through analysis of debriefing interviews and with the use of subjective preference rating questionnaires. The experimental hypotheses will be structured along the following lines in accordance with the research questions:

1. Experimental hypothesis EH_1:
   a. H_0: workload assessment error is less than 10% of baseline
   b. H_1: workload assessment is higher than 10% of baseline

   Independent variable: Workload driver (rest, low, medium, high, very high)

2. Experimental hypothesis EH_2:
   a. H_0: affect assessment error per emotion is less than 10% of baseline
   b. H_1: affect assessment error per emotion is higher than 10% of baseline

   Independent variable: Affect driver (story) at levels of joy, confusion, frustration, boredom, surprise, and anger

3. Experimental hypothesis EH_3:
   a. H_0: adaptive training performance = non-adaptive training performance

b. H_1: adaptive training performance > non-adaptive training performance
Independent variable: without and with adaptive training

In program Year 3 (Option, if funded) we intend to expand the use of UPCAT from individual performance assessment to team performance assessment. However, while there may be some simple cases of team training, we readily acknowledge the complexity of teamwork in general. Well-functioning teams must have a balanced workload with roles that mutually support each other. Unbalanced workload levels may be indicators of dysfunctionality between the team members. In crew resource management research, for example, crew members should have balanced levels of workload and they should both function within their assigned roles rather than having to reach into the other team-member's role. Ellis (Ellis K.E, 2014) measured workload in flight crews and then subjectively assessed each pilot's view of the other's workload. Discrepancies in actual workload and perception of the "other guy's" workload were found to be strong indicators of a team dysfunction. With that in mind, a machine learning feature extraction engine could be added to automatically watch out for such discrepancies.

REFERENCES

Army. (2011a). *The US Army Learning Concept for 2015*. Retrieved from http://www.tradoc.army.mil/tpubs/pams/tp525-8-2.pdf

Army. (2011b). *The US Army Learning Concept, 2012-2020*. Retrieved from http://www.tradoc.army.mil/tpubs/pams/tp525-8-2.pdf

Ellis K.E. (2014). *Fight Deck Crew Coordination Indices of Workload and Situatoin Awareness in Termninal Operations.* (PhD), University of Iowa, Iowa City, Iowa.

Engler, J., & Schnell, T. (2013). *Deterministically Nonlinear Dynamical Classification of Cognitive Workload.* Paper presented at the I/ITSEC 2013, Orlando, FL.

Govindan, R., Narayanan, K., & Gopinathan, M. (1998). On the evidence of deterministic chaos in ECG: Surrogate and predictability analysis. *Chaos, 8*(2), 495-502.

Kozma, R. a. F., W. (2002). Classification of EEG Patterns using Nonlinear Dynamics and Identifying Chaotic Phase Transitions. *Neurocomputing, 44-46*, 1107-1112.

OPL. (2014). *Cognitive Assessment Tool Set (CATS) User Manual* Retrieved from Iowa City, Iowa:

Owis, M., Abou-Zied, A., Youssef, A., & Kadah, Y. (2002). Study of features based on nonlinear dynamical modeling in ECG arrhythmia detection and classification *IEEE Transactions on Biomedical Engineering, 49*(7), 733-736.

Richter, M., & Schreiber, T. (1998). Phase Space Embedding of Electrocardiograms. *Physical Review E, 58*(5), 6392-6398. doi:10.1103/PhysRevE.58.6392

Schnell, T. (2012). *Workload Classification Algorithm Development and Validation for Human Factors Test Vehicle for DIAM*. Retrieved from

Schnell, T., & Engler, J. (2013). *Operator State Sensor Investigations and Operator State Classification* Retrieved from Hampton, VA:

Schnell, T., & Engler, J. (2014). Entropic skill assessment of unmanned aerial systems (UAS) operators. *Journal of Unmanned Vehicle Systems, 2*(02), 53-68.

Schnell, T., Melzer, J. E., & Robbins, S. J. (2009). *The cognitive pilot helmet: enabling pilot-aware smart avionics.* Paper presented at the Head- and Helmet-Mounted Displays XIV: Design and Applications, Orlando, FL, United states.

Sottilare R, Sinatra A, & Boyce M. (2015). *Domain Modeling for Adaptive Training and Education in Support of the US Army Learning Model—Research Outline* Retrieved from

VanLehn K. (2011). The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist, 46*(4), 197-221.

ABOUT THE AUTHORS

Dr. Tom Schnell *is a Professor in the Department of Industrial and Mechanical Engineering at the University of Iowa. He is also the Director of the Operator Performance Laboratory (OPL) where he has been the principal investigator on around 230 research projects. Tom has an undergraduate degree in Electrical Engineering and a MS and PhD in Industrial Engineering. He is a commercial pilot, test pilot, and flight instructor for fixed wing airplanes and rotorcraft.*

Mr. Cory J Reuter *is a full-time Research Assistant at the Operator Performance Lab (OPL). He is currently finishing his undergraduate degree in Mechanical Engineering at the University of Iowa. Before working at OPL, Cory spent several years in the construction industry and has built everything from timber peg homes to designing and building furniture and custom built cabinets. At OPL, he assists in the design and manufacturing of the hardware needed for our human factors research and aircraft maintenance.*

Mr. Ezekiel D. Gunnink *is a full-time Research Assistant at the Operator Performance Lab (OPL). He is currently completing his computer & electrical engineering degree, with a specialization in software engineering. He is an expert in computer vision, graphics, and advancement in sensor technology. He is one of the lead developers and software architects of the CATS software*

Mr. Bradley M. Parker *is a full-time Engineering Coordinator at the Operator Performance Lab (OPL). He is a graduate of Idaho State University where he studied Graphic Arts Technology. He worked in the graphic arts industry for 15 years supporting computer graphics and high resolution imaging systems. Prior to joining OPL, he was an IT support specialist at the University Hospital and Clinics.*

Ms. Jaclyn Hoke *is a Principal Engineer in the Rockwell Collins Advanced Technology Center (ATC) with 10 years of experience in research, technology development, and technical project management. She is the research lead for Rockwell Collins initiatives in State Assessment and Training Effectiveness. Jaclyn has a B.S. in Applied Mathematics, an M.S. in Computer Engineering, and is currently pursuing her Ph.D.*

Dr. Jason D. Moss *is a Research Psychologist for the U.S. Army Research Laboratory (ARL) Human Research and Engineering Directorate (HRED), Advanced Training and Simulation Division (ATSD) in Orlando, Fl. He has over 14 years of experimental research experience in the areas of military psychology, training, perception, human factors psychology, simulator sickness, and virtual environments. He has a Ph.D. in Human Factors Psychology from Clemson University.*

THEME IV: TEAM MODELING

Team Tutoring in the Generalized Intelligent Framework for Tutoring: Current and Future Directions

Anne M. Sinatra[1]
U.S. Army Research Laboratory[1]

INTRODUCTION

While the Generalized Intelligent Framework for Tutoring (GIFT) has been actively developed over the past few years, the majority of projects to date have focused on individual learners. A long-term goal of GIFT is to support team tutoring and provide simultaneous computer-based tutoring at the squad level. Therefore, efforts in the Team Modeling vector have been examining the implications of extending GIFT to team tutoring, and have laid the groundwork for creating team tutors with GIFT.

There are 7 research vectors within the GIFT Adaptive Training project: Learner Modeling, Team Modeling, Domain Modeling, Instructional Management, Authoring, Training Effectiveness and Architecture. In recent years, the Team Modeling vector has branched out from the original Learner Modeling vector. While to date there have been a limited number of projects under the team vector, these projects are beginning to expand, as team tutoring is a goal of GIFT. Further, many of the current and past individual focused research projects could potentially be targets to be adapted or scaled up for team use.

The current paper will (1) discuss the goals of GIFT in regard to team tutoring, (2) discuss the theoretical foundation behind GIFT's team tutoring implementation, (3) discuss the progress made using GIFT for simultaneous two and three player team tutors, and (4) present the future directions and developmental steps being taken toward team tutoring in GIFT. Both the technological and authoring challenges presented by shifting from individual tutoring to team tutoring will be discussed, as well as possible approaches that can be used to help meet these challenges.

GIFT AND TEAM TUTORING

By default, GIFT is set up as a means of providing adaptive tutoring to a single individual who is on his or her own computer. Ultimately, GIFT is intended to be able to handle multiple concurrent users who are being trained on the same material in the same computer-based environment (learners can be either distributed or in the same classroom). Further, GIFT is intended to be able to track the performance of individual team members as well as the team as a whole, and provide adaptive feedback as appropriate. There are two main challenges to being able to conduct team tutoring: (1) technological and (2) authoring. The authoring challenge can be further broken down into the authoring tools required to support team tutoring, and the research/guidelines that should be followed for providing feedback to a team. In regard to the technological aspect, establishing that GIFT messages can be sent as needed between multiple players, a training application, and the GIFT software is the first challenge. In regard to authoring, one of the main challenges is having GIFT's adaptive authoring capabilities set up such that feedback and adaptation rules can be set separately at the individual level based on the specific required tasks, and the team level for overall tasks. Since GIFT is a flexible domain-independent framework, team tutor authoring tools and methods need to be designed to allow for maximum flexibility and combinations of ways to assess the team. The approaches used for authoring team tutors should be able to support teams that have different numbers of members and structures. Therefore, the first authoring challenge is supporting the authoring requirements for team tutoring, and allowing for the flexibility of different domains as well as team configurations. In designing the authoring tools the techniques and approaches for team research and learning should be taken into consideration. There are different strategies that may work better with different types of teams, and GIFT should be able to be configured to support varying types of team taskwork and feedback (e.g., team feedback throughout the training experience vs. after action review team feedback). In GIFT's current structure there is a Domain Knowledge File (DKF) that drives adaptive feedback in a training application (such as Virtual Battlespace 3). The DKF structure needs to be adapted such that rules can be constructed for both teams and individuals in a straightforward, easy to author manner. While some work has begun in this direction, there is no specific authoring tool or authoring plan in place to assist in differentiating between team member roles, and combined team feedback at the current time. In current time, one of the workarounds is using the authoring tools to construct multiple individual DKFs and an overall team DKF which are then all utilized for the interactions. Further, leveraging the adaptive courseflow course object in GIFT with the context of team tutoring has not yet occurred. In the current configuration of GIFT, the adaptive courseflow is run by the eMap and structured by elements of component display theory: rules, examples, recall, and practice (Wang, Goldberg, Tarr, Cintron, & Jiang, 2013). Depending on the performance of the individual learner and characteristics of the individual, different remediation may be provided by the adaptive courseflow. In the current version of the eMap, the rules and examples phases of instruction are largely based on non-interactive static content, supporting the display of file types such as PowerPoints, html, and PDFs. The recall phase is largely multiple choice based, and the practice phase involves interacting with an external training application to demonstrate that the task can be performed by the learner. While remediation on static information that is presented is likely to remain an individual task, the incorporation of team assessment in regard to recall and practice of material could be useful. For example, the recall phase could be updated to allow for multiple team members to work together to answer multiple choice questions, and only submit their responses once everyone has approved them. The practice phase could be updated to support easy authoring of team DKFs for team tutoring and interactions in a training environment.

Team Meta-Analysis and Behavioral Markers

Initial theoretical work to provide the foundation of the team implementation in GIFT was done through a cooperative project with UCF's Institute for Simulation and Training. The project involved a large meta-analysis of team research relevant to team tutoring and the specific goals of GIFT. The initial procedure and results of the meta-analysis were presented at GIFTSym3 (Burke, Feitosa, & Salas, 2015). This project included searching the team literature from the years of 2003 – 2013 for relevant team articles and their outcomes. The meta-analysis identified behavioral, attitudinal, and cognitive contributions in the areas of Team Performance, Team Learning, Team Satisfaction, and Team Viability (Sottilare, Burke, Salas, Sinatra, Johnston, & Gilbert, in review). Additionally, behavioral markers, or indicators of team performance were developed and identified. Markers that were associated with the theoretically identified contributors of trust, collective efficacy, cohesion, communication, and conflict/conflict management were established (Sottilare, Burke, Salas, Sinatra, Johnston, & Gilbert, in review). These markers can be used to assess the team, and help to guide remediation and feedback that they receive. This research serves as the theoretical basis for the team implementation in GIFT. The next steps forward include operationalizing the behavioral markers such that they can be used in a computer-based tutoring environment without the need of a human coder. This project tackles some of the initial team theoretical and authoring challenges of GIFT as it paves the way for identifying the types of markers and measures that should be available to an author who is creating a team tutor.

Initial Team Implementations and Research

The first working team implementations of GIFT have been part of a collaborative effort with Iowa State University. There has been one conducted experiment, and one planned experiment that demonstrates the different functionalities of team research in an intelligent tutoring system environment. The first project is a surveillance task in Virtual Battlespace 2 (VBS2) in which two players work collaboratively to identify and transfer threats that they see in an area that they are monitoring. The first part of this project was to tackle the initial technological challenge of how to get the system to simultaneously monitor two individuals and provide feedback during training. The second piece was designing a team research study that would add to the body of literature about team training in an intelligent tutoring system environment and how to provide team feedback. This task and lessons learned from its implementation have been documented in a number of publications (Gilbert, Winer, Holub, Richardson, Dorneich, & Hoffman, 2015; Bonner, Walton, Dorneich, Gilbert, Winer & Sottilare, 2015; Bonner, et al., 2016).

The output of the studies also has led to considerations about how to grade, assess, and deal with computer-generated team based data (Gilbert et al., in press). A challenge of computer-based team tutoring is being able to have the system assess the team member's performance in real time, and react immediately. The initial study constructed Team DKFs and demonstrated how team data can flow in GIFT. The second study has built upon the original to allow for three team members to engage in the environment simultaneously. In the surveillance task implementation there are individual team member DKFs, and an overall team DKF. This resulted in three DKFs in the original two person task, and will ultimately result in more DKFs as the number of team members increase. For instance, if there is a leader and two additional team members there may be a DKF for the overall team, and each team member, but also potentially for sub-teams. Future work will continue to expand upon team tutoring in GIFT with increasingly more complex teams.

Workshops

In addition to initial theoretical research, workshops have and will be conducted in the future with experts in the areas of team tutoring, team performance, collaborative learning, and team research. The initial workshop titled "Building Intelligent Tutoring Systems for Teams: What Matters" was held at the conclusion of the meta-analysis project in March 2016 in Orlando, FL. Experts discussed the current state of teamwork as it relates to intelligent tutoring systems. The attendees brainstormed about the current and future state of team tutoring, and provided thoughts on steps forward.

The second workshop is the "Team Modeling and Team Taskwork Expert Workshop", which is set to take place in Ames, IA in June 2017. Experts in team research from different backgrounds and focuses will gather to talk about their experiences with team taskwork research, and their ideas on how it can be incorporated into GIFT. This workshop is part of a series that have been conducted since 2012 and will output a book in the Design Recommendations for Intelligent Tutoring Systems series.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

The next step forward in GIFT's team implementation is to expand beyond a two and three person team and into new training environments/tasks. While GIFT has been demonstrated to be capable of handling multiple learners at the same time in a surveillance task, there are more complicated tasks that require hierarchical and varying team roles in order to achieve. Further, future research will focus on operationalizing the behavioral markers that were identified in the initial theoretical research that was conducted. While the identification of these markers was vital, most of them have been primarily established for implementation in an in-person training environment (Sottilare, Burke, Salas, Sinatra, Johnston, & Gilbert, in review). In order to provide opportunities for them to be implemented in real time during computer-based adaptive tutoring, work needs to be put into their operationalization and how they can be authored within GIFT. For instance, if the marker indicated that positive statements between team members led to improved performance, then there needs to be a way for positive statements to be determined by the system, and feedback based on the team using positive statements needs to be authored. While the number of statements made and the meaning could potentially be tackled by semantic analysis, other items like identifying backup behavior (when one teammate is helping another that has fallen behind) may be much more difficult to identify.

Through operationalizing behavioral markers, scaling up the number of individuals who can be involved in a GIFT team tutor, and providing opportunities for more complex teams, GIFT will continue to make strides forward in the area of team tutoring. As these functions continue to develop it will lead to the need for team tutoring authoring tools, and the creation of additional team tutors. The past, current and future efforts will ultimately lead to a straightforward and efficient means of creating team based adaptive tutoring systems using GIFT.

REFERENCES

Bonner, D., Gilbert, S., Dorneich, M. C., Winer, E., Sinatra, A. M., Slavina, A., ... & Holub, J. (2016, September). The Challenges of Building Intelligent Tutoring Systems for Teams. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 60, No. 1, pp. 1981-1985). SAGE Publications.

Bonner, D., Walton, J., Dorneich, M. C., Gilbert, S. B., Winer, E., & Sottilare, R. A. (2015). The Development of a Testbed to Assess an Intelligent Tutoring System for Teams. In *AIED Workshops*.

Burke, C. S., Feitosa, J., & Salas, E. (2015, August). The Unpacking of Team Models in GIFT. In *Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym3)* (p. 49).

Gilbert, S., Sinatra, A. M., MacAllister, A., Kohl, A., Winer, E., Dorneich, M., Slavina, A., Bonner, D., Ouverson, K. (in press). Analyzing Team Training Data: Aspirations for a GIFT Data Analytics Engine. In R. Sottilare (Ed.), *Proceedings of 5th Annual GIFT Users Symposium (GIFTSym5)*: U.S. Army Research Laboratory.

Gilbert, S., Winer, E., Holub, J., Richardson, T., Dorneich, M., & Hoffman, M. (2015, August). Characteristics of a multi-user tutoring architecture. In *Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym3)* (p. 39).

Sottilare, R.S., Burke, S., Salas, E., Sinatra, A., Johnston, J., & Gilbert, S. (in review). Towards a design of process for adaptive instruction of teams: A meta-analysis. In review at Journal of Artificial Intelligence in Education.

Wang-Costello, J., Goldberg, B., Tarr, R.W., Cintron, L.M., & Jiang,H (2013). Creating an advanced pedagogical model to improve intelligent tutoring technologies. In The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

ABOUT THE AUTHORS

Dr. Anne Sinatra *is part of the adaptive training research team within ARL's Learning in Intelligent Tutoring Environments (LITE) Lab, she is lead on the Team Modeling vector and works on the Generalized Intelligent Framework for Tutoring (GIFT) project. Her background is in Human Factors and Cognitive Psychology.*

Analyzing Team Training Data: Aspirations for a GIFT Data Analytics Engine

Stephen B. Gilbert[1], Anne M. Sinatra[2], Anastacia MacAllister[1], Adam Kohl[1], Eliot Winer[1], Michael Dorneich[1], Anna Slavina[1], Desmond Bonner[1] Kaitlyn M. Ouverson[1]
Iowa State University[1], U.S. Army Research Laboratory[2]

INTRODUCTION

The primary questions for any training situation are, "How well did the trainees do? Are they trained enough?" And of course there are secondary questions like, "Who did best/worst?" and "What were their major stumbling blocks?" If the trainees were running a footrace through an obstacle course, these questions would be relatively simple to answer. The measure of time to the finish line answers "how well" and "who did best/worst." The number of stumbles on the obstacles could be counted as literal stumbling blocks. And a threshold based on previous performers could be used to answer whether they are trained enough. Unfortunately, real-world training scenarios are typically much more complex, and answering these simple questions can be quite challenging. This paper discusses some of these challenges, especially in a team training setting.

In the footrace example, if the goal is to train people to run footraces, then the practice footrace is an ideal training environment. More likely, however, the footrace is a proxy for more generalizable skills such as speed and agility. The trainer likely hopes that performance at the footrace will serve as a predictor for performance in real-world scenarios that require those skills (e.g., chasing a suspected terrorist through an urban environment on foot).

In this example, the trainer is essentially attempting to model the learner's skills. The learner's skills at this future real-world scenario are more difficult to measure for several reasons: 1) skills are not directly observable like height and shoe size, 2) skills vary based on mood, motivation, fatigue, and moderating effects of the individual, 3) it is difficult to replicate the real-world scenario for training practice, and 4) relevant real-world scenarios can vary significantly. Thus, a good trainer designs a training experience that will ideally 1) enhance the skills needed for the real-world scenario and 2) provide an accurate prediction of how well the learners will perform on the real-world scenario.

Figure 1 shows a predictive hierarchy of skill measurement that illustrates these ideas with an example. The example shows an attempt to model an individual's communication skills and predict that individual's performance in the battlefield based a virtual training scenario. However, this relatively simple example belies the complexity underlying a team measure. If the trainer wanted to model whether a particular team would excel at communication, individual communication performance measures would need to be combined with additional team performance measures, along with external factors such as the team members' familiarity with each other and their individual levels of experience working on teams.

Researchers and scenario-based trainers have long searched for a systematic method of mapping trainees' behaviors in a technology-based training environment to skill measures. Stacy and Freeman, for example, are addressing this challenge by proposing the Human Performance Markup Language (2016). This paper describes how this challenge is addressed in a team tutor for a surveillance task using GIFT (Sottilare, Brawner, Goldberg, & Holden, 2012) as its tutoring engine. Measures of learning and performance are established by fusing data from the GIFT Event Reporting Tool, VBS2 messaging, and custom scripts that filter data for likely accidental extra keystrokes by participants. We document the assumptions required in this tutor to infer conclusions about learning procedural task skills and abstract team skills from specific behavioral markers.

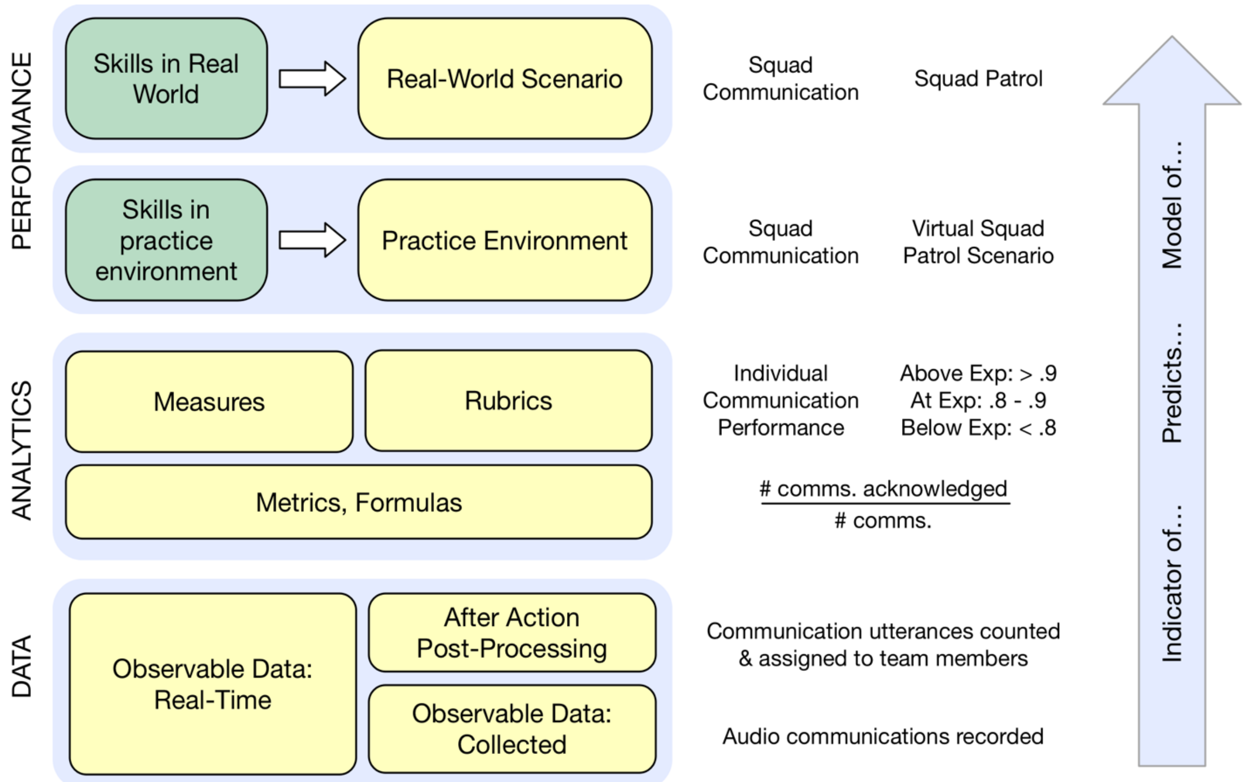


Figure 1: Predictive Hierarchy of Skill Measurement. Concrete observable data at bottom serve as indicators of measures via metrics, rubrics, and formulas. The measures predict skills that can't be directly observed in a practice environment, which is designed to identify skills that will apply in the real world. Many validity assumptions are required. Examples provided at right of hierarchy. Note: for the purpose of this paper, this hierarchy omits self-report data such as surveys.

BACKGROUND: THE SURVEILLANCE TASK

The Surveillance Task is a simple two-person training scenario that was developed as an initial military-relevant testbed to explore intelligent tutoring systems for teams. The primary training objective of the task is building efficient communication behaviors between the two team members. This task uses Virtual Battlespace 2 (VBS2) as the game engine, and GIFT as the tutoring engine. Each team member stands atop a building and conducts surveillance of a 180-degree zone: Member 1 takes the west 180-degree zone, and Member 2 takes the east zone

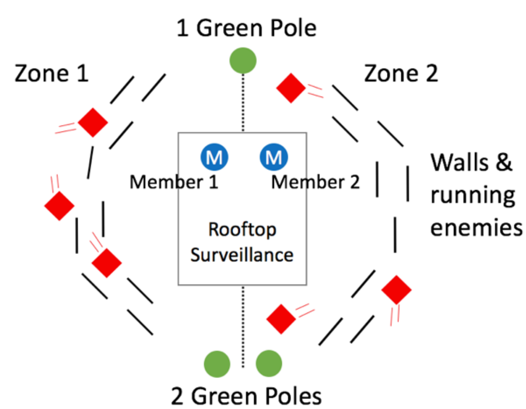


Figure 2: Aerial view of Surveillance Task. Team members (blue M's) alert each other if an enemy (red diamond) moves between zones.

(see Figure 2). During the scenario, enemies (OPFOR) appear from behind walls throughout the environment and move from place to place, sometimes leaving one zone and entering the other. There are two zone boundaries: one by the single green pole, and one by the double green pole. Approximately 50 OPFOR are involved, and the task requires five minutes to complete, growing in difficulty. Participants first watch a 3.5 minute training video, and then a 5-minute practice session. Then they did four consecutive trials.

Figure 3 shows the screen that a learner might see in the Surveillance Task, with a portion of Zone 1 in view at right (along with the single green pole), and GIFT feedback appearing at left. Team members sit in separate closed offices, each with a computer and an open audio channel for communication.

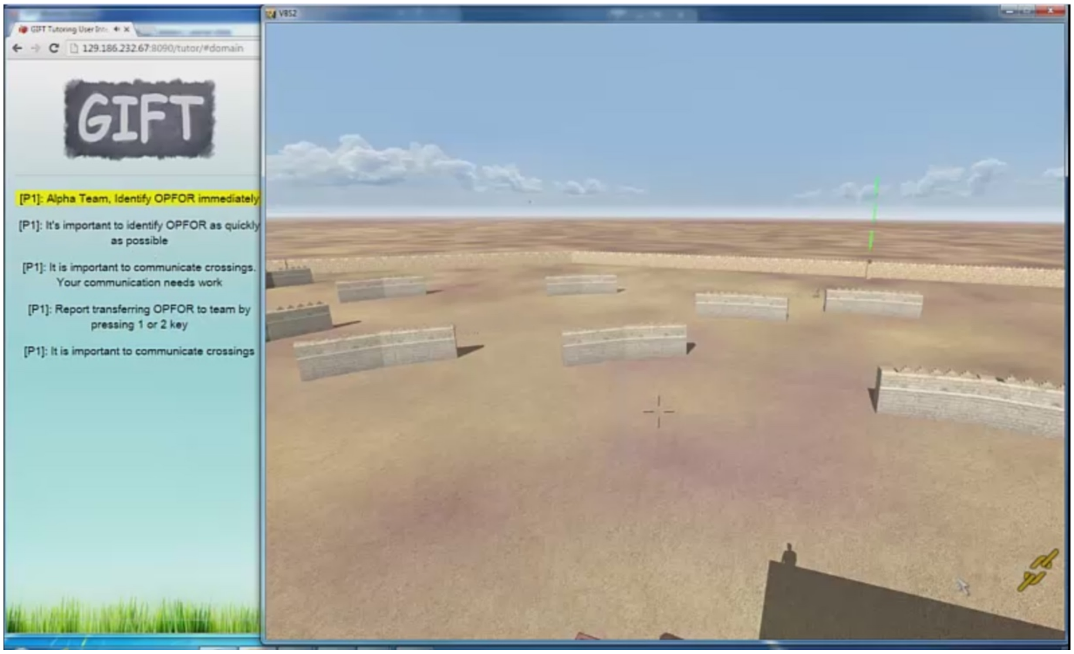


Figure 3: This screenshot of the Surveillance Scenario Tutor shows Team Member 1 looking toward the boundary with one green pole (shown slightly right of center). The team members pan back and forth to observe their full zones. Enemies (OPFOR) run out from behind walls. Team members alert each other if an enemy crosses zone boundaries. GIFT provides feedback at left.

Team members in the Surveillance Task have three duties, or subtasks. Participants' instructions are included below based on the IRB-approved study protocol. The ITS feedback reinforces these instructions.

TRANSFER

- Whenever an enemy entity (OPFOR) is spotted moving towards the edge of your zone, indicate to your teammate that an OPFOR is approaching.
- You must inform your teammate if the OPFOR is approaching from the side with 1 POLE or from the side with 2 POLES.
- You do this by verbally communicating to your teammate as well as pressing the 1 or 2 key, corresponding to the 1 POLE or 2 POLES boundary.
- If there are multiple OPFOR moving towards the same pole you may indicate their number verbally but you must press the appropriate transfer key (1 or 2) for EACH OPFOR.

ACKNOWLEDGE

- Whenever an OPFOR is transferred to you by your teammate you must acknowledge this communication by pressing the E key.
- Transfers should be acknowledged as soon as they are communicated.
- If multiple transfers are communicated, EACH transfer must be acknowledged.

IDENTIFY
- Whenever an OPFOR has transferred from your teammate's zone to your zone, you must indicate that you have seen the OPFOR by pressing SPACEBAR.
- OPFOR entering your zone from the other zone must be identified even if they were not transferred to you by your teammate.
- EACH OPFOR must be identified individually.
- Do not identify OPFOR that have not crossed into your zone yet.

These instructions reveal that the observable data we can gather from participants consists of a keystroke log with time stamps for the keys 1, 2, E, and spacebar, along with recorded audio of verbal utterances. The observable data we can log from the software consists primarily of the location of OPFOR as they run. Evaluation condition code was written for GIFT for each of these subtasks, so that the software can also offer us derivative data of performance evaluations such as Above Expectation, At Expectation, and Below Expectation at any moment in time.

After running participants in this training task, we wanted to ask research questions such as:

- How well did each participant perform with the Transfer, Acknowledge, and Identify subtasks?
- How well did the team of participants do, particularly at communication overall?
- How does each team compare with other teams?
- Did the feedback affect their performance?

To answer any of the these questions, we defined our terms, putting metrics, formulas, measures, and rubrics (ala Figure 1) together to measure constructs such as performance and communication. The remainder of the paper describes this process.

The creation and experimental use of the Surveillance Task has been described in more detail elsewhere (Bonner, Gilbert, et al., 2016; Bonner, Slavina, et al., 2016; Bonner, et al., 2015), but it is worth noting that the Surveillance Task can serve as powerful research platform for exploring the impact of different forms of feedback (textual vs. auditory, positive vs. negative, team-focused vs. individual focused, etc.). Also, the task can be easily scaled in difficulty by adjusting the quantity and timing of the running OPFOR. That said, the platform will only be as powerful as the data analysis available to it, which we explore in the next section.

BACKGROUND: THE EVENT REPORT TOOL WITHIN GIFT

GIFT records all the actions that take place within one of its tutors in log files. The Event Report Tool (ERT) in GIFT was initially developed to be able to extract data from the GIFT logs. As GIFT is a generalized framework that allows for courses to be used for instruction, research, and experiments, some flexibility is built into the ERT. When using GIFT for survey-based experimental data collection, the researcher can select Survey Results and merge files by Username or User ID. However, in order to examine more intricate performance and messages that were recorded in the logs by GIFT, often non-merged individual participant ERT outputs are necessary. When examining data for a single user, working with the ERT and organizing the data in a meaningful way can be challenging. When dealing with team data and multi-player data, it presents an even greater challenge keeping track of which action was done by which team member, and looking for patterns of interactions among team members.

The ERT ultimately needs a way to organize linked data and actions in such a way that a researcher or instructor can easily make sense of it after the fact, a way to create metrics, formulas, and rubrics that can help researchers create data for their measures. By expanding GIFT for teams, the output tools will ultimately need to be adjusted as well. The desktop version is shown in Figure 4. As the tool continues to develop, it would be advantageous to consider adjustments that could support interrelated and team data. In order to analyze data from the current ERT, it is generally necessary to extract and clean the data manually or write code to do so. In terms of Figure 1, the ERT provides the ability to extract the observable data (the lower section), but could benefit from analytics features that would allow the researcher to construct metrics, formulas, rubrics, and measures that could evaluate training constructs like individual performance and team performance. While designing the domain and learner models can be difficult, we suggest that the process of designing the "analysis model" (containing the middle layers of Figure 1) can also be quite time consuming.

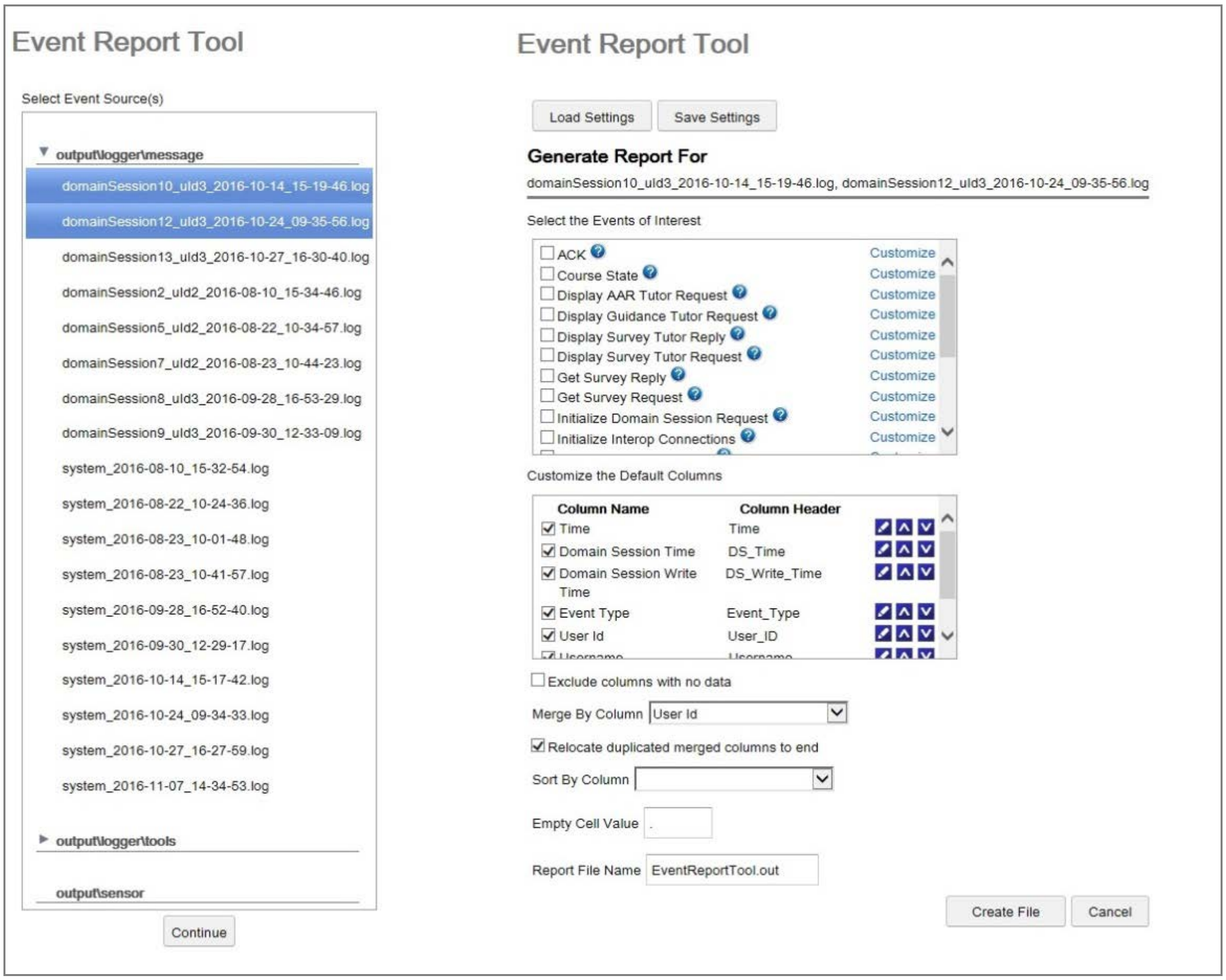


Figure 4. The document selection screen (left) and report generation selection page (right) from the Desktop version of the ERT.

DATA ANALYSIS FROM THE SURVEILLANCE TASK

Table 1 shows some of the measures we wanted to use to analyze the Surveillance Task data and answer the research questions noted above. It is worth noting that while GIFT conditions have been programmed separately to give feedback based on real-time measures of performance, these measures are not always easily convertible to the measures we want for research questions. GIFT's data logging is primarily designed to log data to be post-processed with the ERT. However, the ERT focuses more on data extraction than on enabling data analytics via formulas, metrics, and measures. Therefore, the team developed a custom post-processing system in Python to create data using these analytic approaches.

Table 1: Data Measures, Metrics, & Formulas for Surveillance Task

Construct	Measure	Metric	Formula	Source
Individual Performance	Transfer Rate	Percentage transfers	$\frac{\# \, Transfers}{\# \, OPFOR \; crossings}$	Post-processing
	Acknowledge Rate	Percentage acknowledges	$\frac{\# \, Acknowledges}{\# \, Transfers \; Rec'd}$	Post-processing
	Identify Rate	Percentage Identifies	$\frac{\# \, Identifies}{\# \, OPFOR \; Crossings}$	Post-processing
	Identify Timing	Average time to Identify	$\frac{\sum_i^{Opfor} \lvert ID \; time_i - Trans \; time_i \rvert}{total \; OPFOR \; crossed}$	Post-processing
	Verbal Communication Rate	Percent Verbal Communications	$\frac{\# \, verbal \; comms.}{\# \, comm. keystrokes}$	Behavioral coding of recordings
Team Performance	Team Identify Rate	Total Percentage IDs	$\frac{\# \, Identifies \; from \; both \; players}{\# \, OPFOR \; Crossings}$	Post-processing
	Coordination	Percentage Paired	$\frac{\# \, Trans - Ack \; Pairs}{\# \, Total \; Transfers}$	Post-processing
	Backup Behavior	Percentage IDs w/o Transfer	$\frac{\# \, Identifies \; not \; transferred}{\# \, OPFOR \; Crossings}$	Post-processing
	Team Communication	Communication Count	*# communications total*	Behavioral coding of recordings

After individual participant data files were extracted from ERT in CSV format, they were grouped in team folders. Because each two-person team participated in four trials, there were eight CSV files in each folder. Each team folder could then be imported into the custom data analysis and visualization engine.

At first, post-processing was not a seamless and repeatable process because the ERT did not produce clean interpretable data for analysis. Even though the researcher now had access to information about each player's actions, the formatted CSV files did not provide an understandable representation of the data. They contained a mix of button presses, OPFOR zone states, feedback messages from the tutor, and performance assessments by GIFT. The heart of the custom data analysis engine was a method of structuring the data for each team member to encompass all events needed for metric creation.

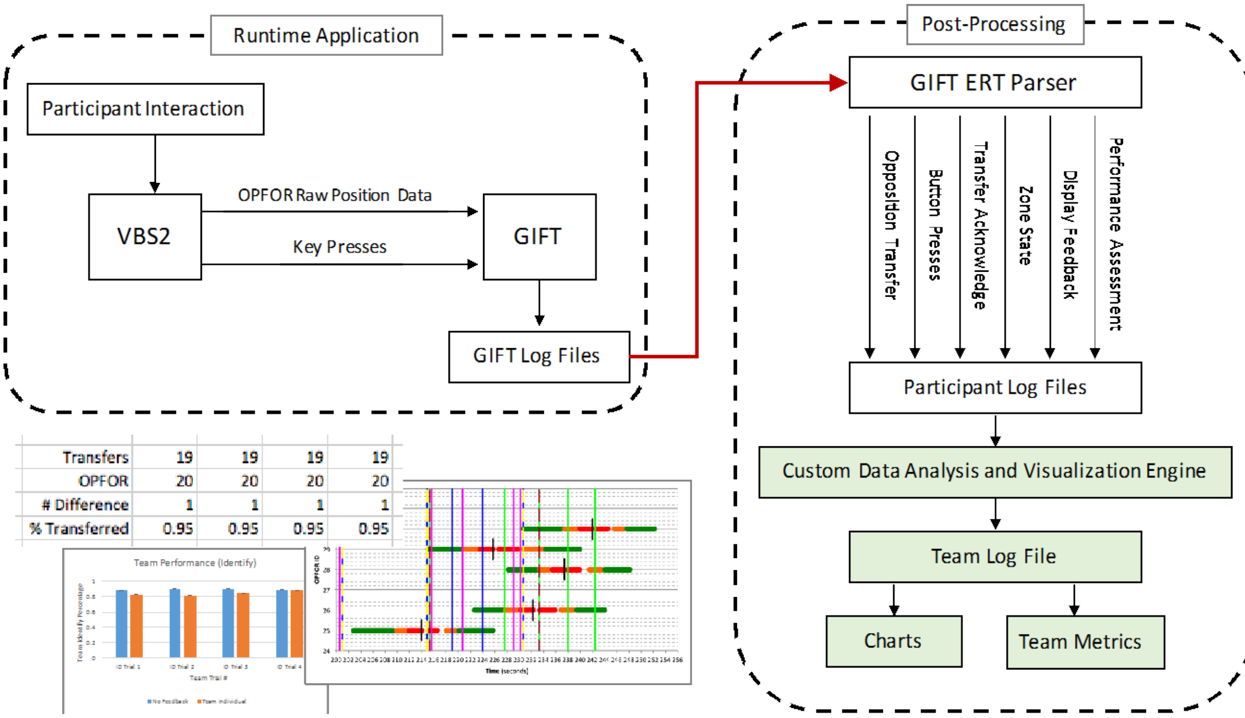


Figure 52: Flowchart of data analysis for the Surveillance Task. Participant key presses are combined with OPFOR position data in GIFT and logged. Data extracted from ERT and processed by a Python-based analysis engine to rearrange data for analysis on a team basis. Green boxes represent custom software development for the Surveillance Task. At lower left are example outputs created for Microsoft Excel.

In Figure 5, first, a team's participant log files were read into the python-based program and parsed by event type. Next, Team Factory class within the software assigned lists of each event type to the team. It's important to note each event contained its own time information, but Team Factory linked them together according to each task. Once parsed and stored, a Metric Manager class created Microsoft Excel spreadsheets containing the metrics and visualizations desired for research analysis. This custom analysis engine save the researchers hours of time by visualizing the data in ways that could be useful to analyze. Additional technical details about the custom data analysis and visualization engine are described elsewhere (MacAllister, et al., 2017).

The Importance of Visualization

As data analysis progressed, it became apparent that the "macro" measures shown in Table 1 were not sufficient to characterize the teams' behavior. Two teams with high Identify Rates, for example, might have very different performance overall. Some teams seemed anecdotally to have different "styles" that were recognizable by the research assistants who ran the participants themselves ("This team communicates a lot" vs. "The team is dominated by one person," etc.), but these patterns or styles were not appearing in the data. The measures in Table 1 were too high in the Predictive Hierarchy shown in Figure 1; we need to see more raw data about the OPFOR themselves. How did they move across the border, and how did the team members react during that movement?

To this end, timeline charts were created for each OPFOR (example shown in Figure 6) that illustrated the zone border crossing process of each OPFOR. Consider that the zone crossing border had a red zone on each side, closest to the exact border, flanked by an orange zone on each side, further from the border, and a green zone on each side, furthest from the border. In the timeline diagram, horizontal bars with those colors are drawn when the OPFOR occupies those zones. The exact border is shown as a short vertical black line between the two red zones. Then, participant actions are indicated by vertical lines. In the ideal performance, there is a Transfer, then an Acknowledge, and then an Identify, all with appropriate timing. When this timing occurred, striped lines were indicated.

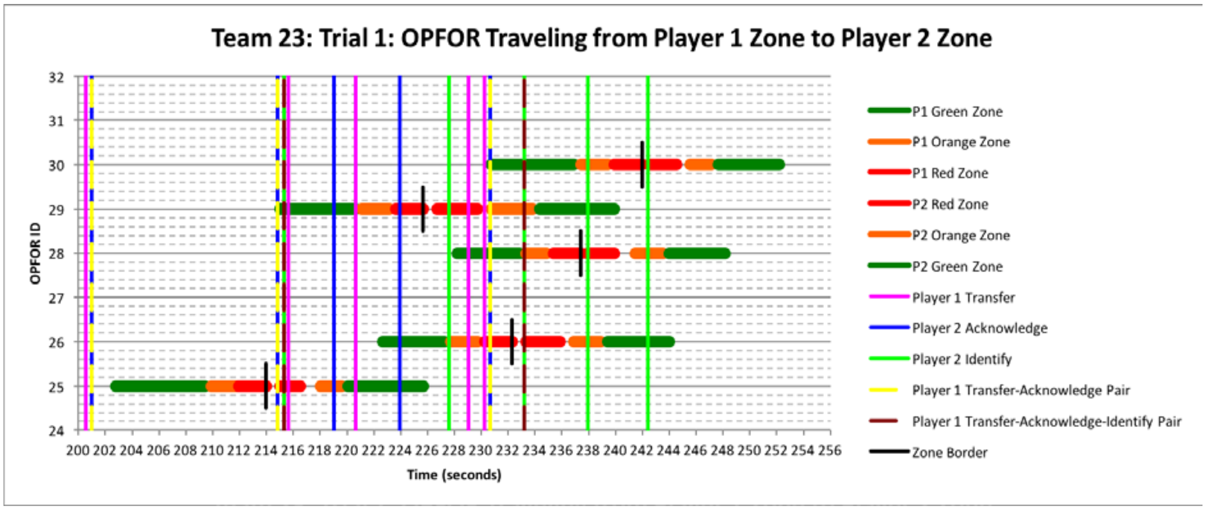


Figure 6: Excerpt of Timeline Chart showing multiple OPFOR paths across the zone borders (horizontal stripes with the border as a short vertical black line). In this figure, five OPFOR cross the boundary at approximately the same time, presenting high cognitive load for participants. Team member Transfer, Acknowledge, and Identify actions are shown as vertical lines. When actions are well-timed, they are designated as "pairs" with striped lines.

It was at this point in the analysis that one of the key difficulties of our Surveillance Task design arose to pose an especially onerous challenge in data analysis. It is not clear, for example, when a group of OPFOR are crossing the border, and a participant indicates *Identify* three times, which Identify action maps to which OPFOR. Sometimes it is obvious from the timing, but oftentimes not, because it is precisely when the task gets stressful with many OPFOR that participants begin to omit actions. Thus, it can be difficult to map actions to OPFOR when there are, say, five OPFOR and only three Identifiers. Or, if a participant is particularly prone to accidentally pressing keys multiple times when only one press is intended, e.g., during the heat of a high cognitive load task, should the parser flag the later presses as extra or the initial presses?

To address these challenges, the research team developed Data Analysis, Labeling & Interpretation (DALI) Rules for these timeline diagrams. Using these rules, three human labelers were asked to assign every vertical line to an OPFOR or mark it as extra for 10% of the data. Once interrater reliability was established, one rater continued to mark the rest of the data. The DALI rules had clarifications of how to resolve ambiguities, like "If you have 2 Green lines, as long as they're both after the pink and blue lines, accept the first green and mark the 2nd green as extra." This approach led to hand labeling as shown in Figure 7. Once the labels were placed, data could be added to an Excel sheet, and further automated processing could take place.

The research team was disappointed that manual labelling was required, and briefly attempted to automate the process. However, it was quickly discovered that the ambiguities of a border crowded with OPFOR led to such complicated software rules that it would actually be faster for the team the develop the DALI rules and manually label all 136 stripe diagrams (including some duplicate labeling by multiple raters to ensure good interrater reliability) than it would be to develop and carefully test software rules.

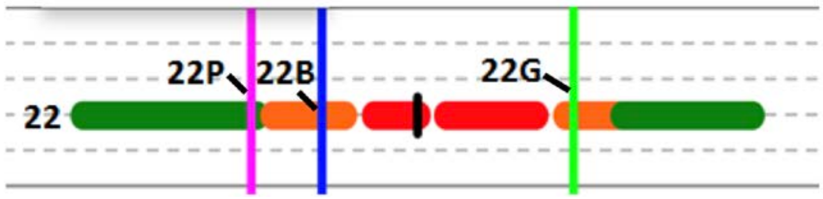


Figure 7: Hand-labeled pink, blue, and green vertical lines to map them to each OPFOR.

Data analysis based on timelines and DALI rules is currently underway, and will ideally lead to better understanding of the team behavior. These more detailed data will lead to team metrics such as **coordination timing** (how quickly one teammate responds to the other), **coordination symmetry** (whether each teammate responds equally quickly to the other), **team style** (unique transfer-acknowledge-identify timing patterns that we have noticed anecdotally can make teams identifiable), and **team cognitive capacity** (at what point are there too many OPFOR for a team). These more detailed metrics will be much more informative than the Table 1 Macro Metrics. We look forward to describing these results in a future paper.

CONCLUSIONS AND RECOMMENDATIONS FOR GIFT

This case study of data analysis of a team tutor illustrates the significant challenges of analyzing complex team data, even for what seems like a relatively simple task from the participants' perspective. As demonstrated, the full spectrum of predictive data analytics and interpretation (ala Figure 1) was needed to evaluate the result of the Surveillance Task. The authors conclude that a system for complete end-to-end assessment of a team's team skills and task skills based on members' performance in a simulation will indeed need to draw on the full predictive hierarchy of skill measurement. The specific elements in the hierarchy may differ by team scenario, but elements at each of the three main levels of the hierarchy (data, analytics and performance) will need to be present for team assessment.

As described above, GIFT currently has no tools for data analytics and visualization. This lack could point to a future vision of a GIFT InfoVis module, or to not reinvent the wheel, perhaps GIFT could create APIs that allow easy movement of data to Tableau, R, and other visualization tools. In addition, as described in the section about the ERT, it will be critical for team tutoring in the future for GIFT to allow data from multiple team members to be affiliated for easy analysis.

REFERENCES

Bonner, D., Gilbert, S., Dorneich, M. C., Winer, E., Sinatra, A. M., Slavina, A., et al. (2016). *The Challenges of Building Intelligent Tutoring Systems for Teams.* Paper presented at the Human Factors & Ergonomics Society (HFES) Annual Meeting, Washington, D.C.

Bonner, D., Slavina, A., MacAllister, A., Holub, J., Gilbert, S., Sinatra, A. M., et al. (2016). The Hidden Challenges of Team Tutor Development. In R. Sottilare & S. Ososky (Eds.), *Proceedings of 4th Annual GIFT Users Symposium (GIFTSym4)* (pp. 49-60): U.S. Army Research Laboratory.

Bonner, D., Walton, J., Dorneich, M. C., Gilbert, S. B., Winer, E., & Sottilare, R. A. (2015). The Development of a Testbed to Assess an Intelligent Tutoring System for Teams. In *Proceedings of the Workshops at AIED 2015*. Madrid, Spain.

MacAllister, A., Kohl, A., Gilbert, S., Winer, E., Dorneich, M., Bonner, D., et al. (2017). *Analysis of Team Tutoring Training Data*. Paper presented at the MODSIM World 2017, Virginia Beach, VA.

Sottilare, R., Brawner, K. W., Goldberg, B. S., & Holden, H. K. (2012). The Generalized Intelligent Framework for Tutoring (GIFT).

Stacy, W., & Freeman, J. (2016). Training objective packages: enhancing the effectiveness of experiential training. *Theoretical Issues in Ergonomics Science*, *17*(2), 149-168.

ABOUT THE AUTHORS

Dr. Stephen B. Gilbert, *is an associate director of the Virtual Reality Applications Center and assistant professor of Industrial and Manufacturing Systems Engineering at Iowa State University. His research interests focus on technology to advance cognition, including interface design, intelligent tutoring systems, and cognitive engineering. He is a member of IEEE and ACM and works closely with industry and federal agencies on research contracts. He is currently PI on a project supporting the U.S. Army Research Laboratory STTC in future training technologies for teams.*

Dr. Anne M. Sinatra *is an Adaptive Tutoring Scientist and Research Psychologist at U.S. Army Research Laboratory. She is the team modeling vector lead on the GIFT project's Adaptive Training Team. Her background is in human factors psychology, cognitive psychology, and education.*

Anastacia MacAllister, MS *is a PhD student in Mechanical Engineering and Human-Computer Interaction at Iowa State University's Virtual Reality Applications Center. She is working on developing Augmented Reality work instructions for complex assembly and intelligent team tutoring systems.*

Adam Kohl *is a MS student in Mechanical Engineering and Human-Computer Interaction at Iowa State University's Virtual Reality Applications Center. He is working on developing pattern recognition techniques to enhance n-dimensional data visualization methods.*

Dr. Michael C. Dorneich *is an Associate Professor at Iowa State University in the Industrial and Manufacturing Systems Engineering Department. His background in in human factors and cognitive engineering, and his research interests focus on creating adaptive human-machine systems in intelligent tutoring, automation, and displays.*

Dr. Eliot Winer *is an associate director of the Virtual Reality Applications Center and professor of Mechanical Engineering and Electrical and Computer Engineering at Iowa State University. He is currently co-leading an effort to develop a next-generation mixed-reality virtual and constructive training environment for the U.S. Army. Dr. Winer has over 15 years of experience working in virtual reality and 3D computer graphics technologies on sponsored projects for the Department of Defense, Air Force Office of Scientific Research, Department of the Army, National Science Foundation, Department of Agriculture, Boeing, and John Deere.*

Desmond Bonner, MS, *is a PhD student in Human Computer Interaction and Industrial Engineering at Iowa State University. He serves as an assistant researcher on the Adaptive Team Training Project. His background is in interactive design.*

Anna Slavina, MS, *is a Ph.D. student in Psychology, Education, and Human-Computer Interaction at Iowa State University's Virtual Reality Applications Center. Her research interests include the effects of technology on memory, intelligent tutoring systems, and visually induced motion sickness.*

Kaitlyn M Ouverson *is a MS student in Human-Computer Interaction at Iowa State University's Virtual Reality Applications Center. Her background in experimental, social and cognitive psychology drives her interest in team facilitation in the virtual world, the real world, and any mix of the two.*

THEME V: INSTRUCTIONAL MANAGEMENT

Pedagogical Management in Support of a Generalized Framework for Intelligent Tutoring

Benjamin Goldberg[1]
U.S. Army Research Laboratory[1]

INTRODUCTION

A current goal associated with the development of the Generalized Intelligent Framework for Tutoring (GIFT) is providing a set of tools for training practitioners to rapidly build adaptive instructional materials based on an interplay of knowledge acquisition and skill development. To accommodate this guiding requirement, an instructional management research vector was devised as a means to coordinate resources and efforts to meet the needs of end users. While the science surrounding intelligent tutoring system (ITS) development is multidisciplinary, a guiding assumption is that targeted GIFT developers in the military community will be subject matter experts (SMEs) within their respected fields, but lack many of the technical disciplines that go into ITS development. Accordingly, it is also safe to assume these SME developers will also require assistance in authoring training content that adheres to learning science principles. As such, authoring workflows and ITS methods must be developed to compensate for the skills a GIFT user lacks when creating a lesson or course.

Due to this challenge, GIFT development in the instructional management vector aims at providing a means for embedding pedagogical theory into GIFT schemas and authoring workflows. The goal is to develop enabling technologies that allow SMEs to author GIFT-based lesson materials that are empirically informed and grounded in instructional design theory. In this paper, we present the current state of GIFT as it relates to instructional management capabilities and associated research to extend how GIFT can manage and personalize training interactions. As an organizing function, we arrange GIFT pedagogy into three temporal categories: (1) instructional management at the lesson level, which personalizes content and adapts course sequencing based on performance and persistent learner attributes (i.e., outer-loop adaptation), (2) instructional management at the interaction level, which deals with real-time coaching and scenario manipulation across an array of practice events (i.e., inner-loop adaptation), and (3) instructional management at the after-action review level, which focuses on reflection and remediation practices following completion of a learning event. Each category has a set of unique features dictated by instructional theory, with on-going projects informing their design and evaluating their utility. We will present the foundations informing the methods applied, and the current state of their practice. We will conclude with future directions of GIFT development, and how the instructional management research vector is aligned to meet training requirements road mapped for future training applications and methods.

Dimensions of Instructional Management Research

In November 2015 members of the GIFT team published a research outline that examined specific goals and interests associated with instructional management in ITS type environments (Goldberg, Sinatra, Sottilare, Moss & Graesser, 2015; Goldberg, Sottilare, Moss & Sinatra, 2015). The authors identified the following dimensions as critical benchmarks in driving capability enhancements to GIFT pedagogical practices:

- Guidance and Scaffolding: focuses on identifying a set of pedagogical best practices that adhere to the tenets of learning and skill development. The challenge is identifying methods that generalize across domains and task environments, and providing tools flexible enough to create scaffolding that can be represented in domain-agnostic terms. Current research aims at creating logic to manage timing, specificity, and modality determinations of intervention content at the individual level.

- Social Dynamics and Virtual Humans: focuses on the social component of learning, and building tools and methods that adhere to the social cognitive tenets of how individuals interact to instill knowledge and solve problems (Bandura, 1986; Vygotsky, 1978). From an adaptive instructional management standpoint, social dynamics is concerned with: (1) using technology to replicate interactive discourses common in learning and operational settings, (2) using technology to create realistic and reactive virtual humans as training elements in a simulation or scenario, and (3) using technology to create social networks for the purpose of supporting peer-to-peer and collaborative learning opportunities.

- Metacognition and Self-Regulated Learning (SRL): focuses on instructional management practices that aim at building habits linked to successful regulation of learning practices and that promote metacognitive applications. This approach to instructional management varies from traditional guidance and scaffolding techniques as it focuses on behavior and application of strategy, rather than on task dependent performance. This research area is of interest as it is based around GIFT supporting SRL, and the efficacy of defining and modeling persistent metacognitive strategies that can be applied across domain applications. The goal is to embed instructional supports that promote situational awareness, and guide learners in planning, monitoring, and reflection based activities.

- Personalization (Occupational and Non-Cognitive Factors): focuses on the use of learner dependent information to personalize a training experience. This can involve personalizing content based on interests, with the goal of inducing a higher level of motivation when the context of a learning event is framed within a use case the learner cares about. In addition, the personalization dimension is also interested in identifying ways to automatically personalize training interactions based on occupational factors that are unique to their upcoming assignment or current job description. All of these instructional management practices require research to identify mechanisms for easily implementing personalization techniques, along with empirical evidence supporting their application for wide GIFT application.

The dimensions reviewed above provide a means for organizing and prioritizing efforts to enhance GIFT's current instructional management support. While the research outline mapped out desired end-state functions of GIFT, it is also important to capture the current state of practice, as those piece parts are the ultimate methods rolled out to the community at large. In the remainder of this paper, we identify GIFT's instructional management functions, and how they apply to future enhancements that aim to meet the goals of the overarching instructional management capability dimensions.

CURRENT PRACTICE OF INSTRUCTIONAL MANAGEMENT IN GIFT

In the remainder of the paper, we present the current state of practice as it pertains to instructional management capabilities built within GIFT. Much of the projects over the past year have been influenced by the road mapping exercise captured in the research outline described above. GIFT pedagogy and instructional management will be described at three levels of interaction, each with a distinct set of adaptive options supported. These include: (1) the lesson level, (2) the interaction level, and (3) the after-action level.

Instructional Management at the Lesson Level

At the lesson level GIFT provides the tools to build a sequence of course objects that dictate the learner experience (see Figure 1). Course objects are designed to either inform the learner, collect information from the learner, or manage execution of content delivery and assessment across problem-sets and scenarios configured during the authoring process. From an instructional management standpoint, pedagogy at the lesson level focuses primarily on lesson sequencing and personalization. It manages adaptations at the macro outer-loop level, where models have been established that dictate what a learner will experience next based on an established learner model and performance data captured during prior interactions. The lesson level pedagogy logic is currently captured within GIFT's first generalized pedagogical model called the Engine for Management of Adaptive Pedagogy (EMAP).

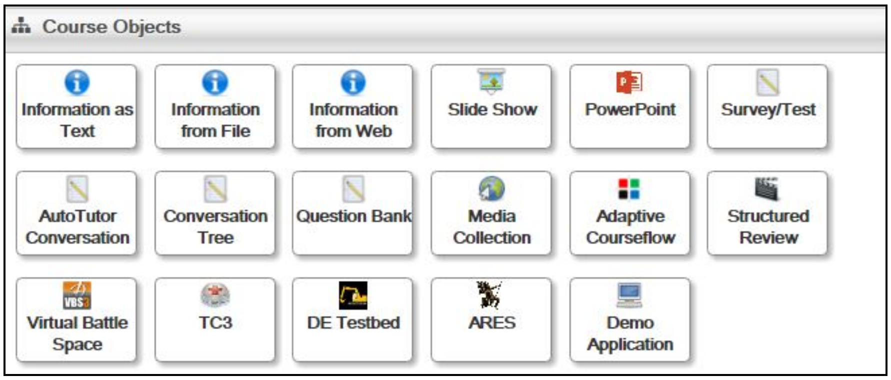


Figure 1. Current list of supported GIFT course objects

EMAP

The EMAP is GIFT's first domain-independent pedagogical model informed by instructional theory, with much of the research and development documented over the years (Goldberg et al., 2012; Wang et al., 2013; Goldberg, Hoffman & Tarr, 2015). What the EMAP provides is a theoretical instructional design framework embedded in GIFT that guides the authoring and configuration of adaptive learning experiences. The EMAP is based on David Merrill's component display theory (CDT), with learning broken up into four primary categories: (1) presenting rules of a domain, (2) presenting examples of those rules applied, (3) asking the learner to recall information as it relates to the domain, and (4) allowing the learner to practice the application of those rules in a novel context for the purpose of skill development (see Figure 2).

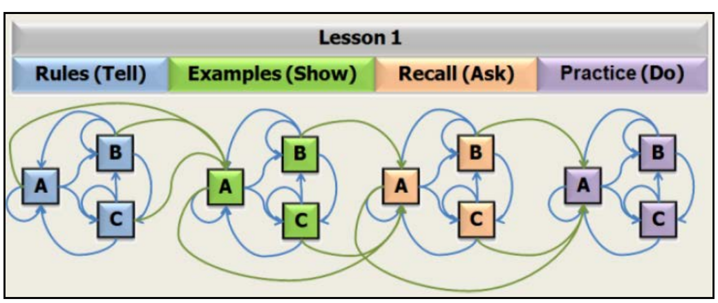


Figure 2. EMAP representation for a lesson teaching three overall concepts and all possible permutations based on variations in assessment outcomes

These four categories are the building blocks of the Adaptive Courseflow GIFT course object. Each category is embedded within the object's schema, where an author configures the content, assessments, and practice events in each categorical bin. This includes loading in all relevant lesson content into the rules and examples bins, and establishing metadata tags for each file. The metadata tags are currently informed from IEEE's Learning Object Metadata (LOM) standard (Mitchell & Farha, 2007), and are used to describe what that associated piece of content covers and the type of materials established within (e.g., videos, figures, worked examples, etc.). GIFT has back end logic informed from a populated pedagogical configuration file that matches learner model attributes (e.g., prior knowledge, motivation, etc.) with metadata descriptors that associate with content, difficulty, and scaffolding type recommendations.

The resulting decision tree was created following the completion of a literature review that aimed to capture instructional management best practices that could be organized at a domain-independent level (Goldberg et al., 2012). A recognized limitation from the literature review was a lack of generalized findings across numerous studies. This is due in part to the nature in which instructional strategies are defined. Each generalized strategy must be contextualized into the domain and application context it will be delivered within, thus producing confounding factors for defining the specific characteristics associated with an investigated intervention. However, a substantial finding from this project was the recognition of four learner attributes found to account for consistent variance in performance outcomes across studies, including (1) prior knowledge/skill, (2) motivation, (3) self-regulatory ability, and (4) grit/perseverance (Wang et al., 2013). These variables served as the moderators to base the first EMAP instantiation around. Access the GIFT documentation for a complete breakdown of the EMAP and its underlying logic:

(https://gifttutoring.org/projects/gift/wiki/Engine_For_Management_of_Adaptive_Pedagogy_(eMAP)_2020-1).

A recognized limitation of the current EMAP is the deterministic nature in which the model was developed. The EMAP functions as a decision tree that maps learner traits and attributes with content descriptors through GIFT's pedagogical configuration file. The configuration file is referenced at runtime, with a content selection algorithm in place that identifies the best piece of content based on concept coverage and the most metadata matches based on a learners profile and available rule and example content. In addition, the remediation logic was also recognized as being rather simplistic, where it would select a new piece of content, if one existed, covering the concept not meeting assessment criteria in either the recall or practice phase of the EMAP interaction. In these instances, the remediation logic passes a learner back into the rule or example category where content is selected for presentation. This remediation approach assumes there is material designated to support an intervention and that the learner has the required understanding to use the new information to correct the misconceptions or impasses identified during assessment.

EMAP Enhancements

A current effort led by North Carolina State University and Intelligent Automation, Inc. is applying methods grounded in tutorial planning and reinforcement learning to extend the current state of the EMAP to support a stochastic modeling approach that introduces probabilistic reasoning in the selection of tutorial actions carried out by GIFT at the lesson level. The approach incorporates embedding a learning activity framework put forth by Chi (2009) that differentiates activities undertaken by learners into constructive, active, and passive (CAP) interactions. This CAP activity framework enables the incorporation of multiple remediation types for a given concept, and is well-suited for markov decision processes (MDPs) to dictate what intervention is best for what type of learner. In addition, this approach supports a back-end reinforcement learning method that trains and optimizes the MDP policies over time as more data and evidence is made available following interactions from a large set of learners.

To accommodate this approach, the adaptive courseflow GIFT course object is being re-factored to support a category of content designated for remediation purposes. Now, an author has the ability to establish core rule and example lesson materials (i.e., content all learners will see regardless of learner profiles), with a remedial bin that enables a developer to build CAP-based tutorial interactions for remediation support following assessment events managed by the EMAP. When performance states are available, the EMAP will select remedial materials based on outputs from the MDP policies. To support policy optimization, a GIFT reinforcement learning tool is being developed for the purpose of updating policies as evidence is made available on the utility of their application. The described enhancements are currently under development during the writing of this chapter, with data collections planned for model training and evaluation over the next twelve months.

EMAP and Massive Open Online Courses (MOOCs)

With the EMAP providing a generalized pedagogical framework to structure personalized content selection and remediation, one area of interest is how GIFT fits within the context of MOOCs. A current effort in collaboration with the University of Pennsylvania and Carnegie Mellon University is looking at the utility of GIFT in providing a standardized framework for personalizing MOOC interactions. For the effort, GIFT is being integrated with the MOOC platform edX (https://www.edx.org/). By making GIFT compliant with the Learning Tools Interoperability (LTI; Severance, Hanss & Hardin, 2010) standard, developers in edX can reference GIFT configured lessons within the courseflow of their MOOC. In this instance, a MOOC can handover control to GIFT for a designated lesson where the EMAP can be applied to personalize the experience an individual receives based on the content and assessment made available by the lesson creator. Following completion of that lesson, GIFT can communicate results and behavior data back to edX for performance tracking and accreditation purposes. These described features are currently being developed, with experimentation planned to determine if personalization practices improve MOOC usage and overall learning outcomes.

Instructional Management at the Interaction Level

At next level of GIFT pedagogy, instructional management has to do with monitoring real-time interaction and managing specific events through coaching and scenario adaptation practices. A major function of GIFT is its ability to capture real-time interaction data from external training environments through an interop configuration. These environments are used to support experiential practice type opportunities for learners, and range from PowerPoint slide presentations to interactive first-person shooter type game environments. With an established gateway module, GIFT can capture data produced from any external system and route specified information into the domain module for assessment purposes (Sottilare, Goldberg, Brawner & Holden, 2012). In the domain module, a domain knowledge file (DKF) is configured for the purpose of contextualizing raw system data around a set of concepts represented in a task ontology. The ontology organizes a scenario into a set of tasks a learner will be asked to perform, a set of concepts for each task a learner will be measured against, and a set of conditions defined to inform the performance measurement of those concepts. In this instance, raw data is used for the purpose of informing measures to gauge performance and infer competency.

From an instructional management perspective, there are currently four supported pedagogical requests communicated from the pedagogical module to the domain module when interacting with an external training application. These include, (1) request-instructional-intervention (e.g., provide guidance through the form of a hint or prompt; see Figure 3 for an example of a coaching hint communicated by GIFT), (2) request performance assessment (e.g., ask a question of the learner to update learner state), (3) request scenario adaptation (e.g., modify the scenario or problem to adjust difficulty), and (4) do nothing. These message types are currently informed by observable performance state transitions across the concepts being tracked in a DKF (e.g., performance on concept 2a transitioned from at-expectation to below-expectation). These transitions are reported out to the learner module, where the learner state is defined, including performance and all other relevant attributes in the learner model. The learner state is then passed to the pedagogical module for determining intervention type at the individual level.

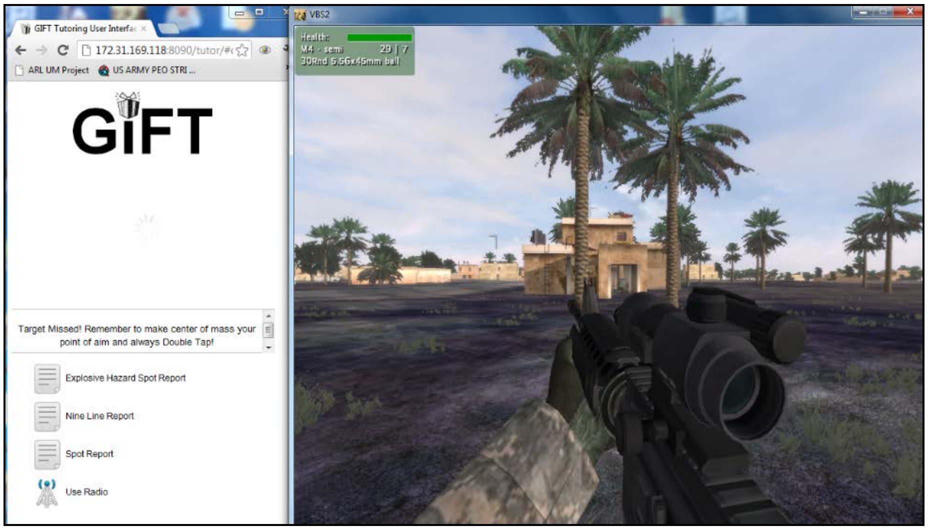


Figure 3. GIFT providing a real-time hint on 'Rules of Engagement' concept in a Virtual Battle Space scenario

In its current state, a GIFT developer configures the strategy types to be enacted when a specific concept transition is observed. The developer is then responsible for translating that strategy request into a specific tactic to be executed at run-time (i.e., define the exact hint to be presented when a request-instructional-intervention request is received for any given concept in the DKF). While the tools and methods in GIFT for interaction level pedagogy support real-time interventions when performance conditions are met, there is much work left to be done in determining how best to intervene and adapt based on individual differences and learner profile information.

DKF Enhancements to Support CAP Tutorial Planning

A recognized limitation in the current run-time instructional support of GIFT for individuals interacting with an external training application is the lack of logic to drive personalized coaching based on individual differences. At the moment, regardless of information associated with an individual's learner model, GIFT is set up to provide the same real-time tutorial actions based on observed transitions defined in the DKF. In lieu of this technical short-fall, current work is investigating the application of the aforementioned CAP instructional activity model to fit within the context of DKF pedagogical practice. This will enable GIFT to provide passive feedback information during a training event when appropriate, while also enabling GIFT to intervene with targeted activity exercises that are aimed at coaching a specific concept or skill when assessment logic (i.e., tutorial planning MDPs) deems it appropriate. The policies are designed to optimize overtime through a reinforcement learning method applied to the back-end as data is made available across practice scenarios. This also supports a reuse philosophy of intervention materials, where CAP associated activities can be triggered at both the lesson and interaction level, reducing the requirement to author interventions across both instances.

Metacognitive Training

Another area of instructional management the GIFT system aims to support is using pedagogical practices to aid learners in developing metacognitive skills that associate with self-regulated learning, critical thinking, and on-the-spot problem solving (Biswas, Segedy & Kinnebrew, 2014). For this reason, a current effort in collaboration with Vanderbilt University is investigating how to use GIFT to model self-regulated learning behaviors and metacognitive skills for the purpose of driving interventions aimed at improving how individuals go about solving a problem, rather than focusing on the problem solution itself. While the main component of this research is focused on the learner modeling aspect to start, the overarching theme of the project is driven by eventual instructional support currently not provided in GIFT. The resulting effort is restructuring the hierarchical schematics of the DKF to support a layered inference procedure (see Rajendran et al.'s technical discussion on the learner modeling work further down in the proceedings). With a learner modeling approach in place, GIFT can infer an individual's understanding of metacognition and self-regulated learning for the purpose of driving focused interventions that guide users in the application a identified behaviors congruent with effective problem-solving applications.

Psychomotor Skill Development and Coaching

Another exciting area of instructional management research is seen in examining GIFT's utility to train psychomotor skill domains. This is a complicated application of ITS as it breaks away from common cognitive problem spaces these systems are traditionally developed within. As such, there is still much research to be done on the modeling and pedagogy components of training a psychomotor task in the absence of human instructors. From an instructional management perspective, work is being performed to account for psychomotor training at both the lesson and interaction level (see Brown, Bell & Goldberg paper in the proceedings for a full breakdown of conceptualized approach being implemented). At the lesson level, this involves creating course objects based on the abstraction of the EMAP that takes into account instructional theory models grounded in psychomotor application. This breaks away from the EMAP's dependency on the CDT, where these objects can now associate with any number of instructional models that a developer wants to base their interactions within. At the interaction level, research is required to determine how best to manage feedback and remediation practices. This involves creating data capture techniques that provide inputs used for modeling physical behaviors, and building representations of those behaviors that are used to guide assessment practices that ultimately inform pedagogical decisions.

Instructional Management and After Action Review

While much of the current instructional management functions in GIFT focus on real-time interactions at varying levels of granularity, new tools and methods are being developed to support personalized after-action review (AAR) materials. For complex skill domains, AARs serve as critical functions in the training process. In these instances, learners have the opportunity to reflect on problems and scenarios undertaken for the purpose of critiquing their own interaction and understanding the implications of their actions on reaching scenario and specific task objectives. A current effort (see Carlin, Brawner, Nucci, Kramer & Oster in the proceedings for details on AAR approaches being investigated in GIFT) is examining how to apply modeling methods to create individualized AAR interactions based on what is observed across a GIFT managed lesson. The goal is to develop technologies that automatically identify critical errors and misconceptions by the learner(s) and automatically select an optimal instructional path and associated instructional content to construct an AAR. The project is applying MDP inference procedures for identifying concepts to personalize an AAR around, along with the organization of content and activities that target the goals of the AAR interaction. These goals include reinforcing learning objectives, addressing impasses, and contextualizing the lesson and training with real world application through mental reflection exercises.

CONCLUSIONS

In this paper, we presented a snapshot of current instructional management capabilities within GIFT, along with ongoing efforts aimed at enhancing applied methods. GIFT is a moving target in terms of development, so it is important to document the methods applied within GIFT and the research that went into its implementation. The pedagogical infrastructure in GIFT is maintained at the lesson level, the interaction level, and the after action level where varying modeling techniques are applied to determine the instructional adaptation/intervention to enact. There is still much work to be done before GIFT's pedagogical practices are easily implemented in an operational context. In addition, GIFT must be able to adapt pedagogical practice as future training instances and applications are developed and transitioned to the Warfighter.

Future Directions

Current trends in ITS research as it relates to the GIFT project is focusing on two fronts: (1) using adaptive training and education practices to support team development and cohesion and (2) using adaptive training and education practices within mobile applications to support ease of access and on-the-spot training support. These themes will be addressed in the coming years as team-based and mobile-based ITS applications mature. Both themes are being addressed in current projects, but the instructional management components associated with their instantiations have yet to be examined. As the future of training and education evolves, GIFT is set up to instructionally support all facets of learning and skill development.

REFERENCES

Bandura, A. (1986). *Social Foundations of Thought and Action: A Social Cognitive Theory*. Englewood Cliffs, N.J.: Prentice Hall.

Biswas, G., Segedy, J. R., & Kinnebrew, J. S. (2014). A Combined Theory-and Data-Driven Approach for Interpreting Learners' Metacognitive Behaviors in Open-Ended Tutoring Environments. In R. Sottilare, A. Graesser, X. Hu & B. Goldberg (Eds.), *Design Recommendations for Intelligent Tutoring Systems (Volume 2: Instructional Management)*.

Chi, M. T. (2009). Active-constructive-interactive: A conceptual framework for differentiating learning activities. *Topics in Cognitive Science*, 1(1), 73-105.

Goldberg, B., Brawner, K. W., Sottilare, R., Tarr, R., Billings, D. R., & Malone, N. (2012). 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.

Goldberg, B., Hoffman, M., & Tarr, R. (2015). 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.

Goldberg, B., Sinatra, A., Sottilare, R., Moss, J., & Graesser, A. (2015). Instructional Management for Adaptive Training and Education in Support of the US Army Learning Model-Research Outline: DTIC Document.

Goldberg, B., Sottilare, R., Moss, J., & Sinatra, A. (2015). *Dimensions of Instructional Management Research in Support of the Generalized Intelligent Framework for Tutoring (GIFT).* Paper presented at the Defense and Homeland Security Simulation (DHSS) Workshop, Bergeggi, Italy.

Mitchell, J. L. & Farha, N. (2007). 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.

Severance, C., Hanss, T., & Hardin, J. (2010). Ims learning tools interoperability: Enabling a mash-up approach to teaching and learning tools. *Technology, Instruction, Cognition and Learning, 7*(3-4), 245-262.

Sottilare, R., Goldberg, B., Brawner, K. W., & Holden, H. (2012). *Modular Framework to Support the Authoring and Assessment of Adaptive Computer-Based Tutoring Systems*. Paper presented at the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC), Orlando, FL.

Vygotsky, L. S. (1978). *Mind in Society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.

Wang-Costello, J., Tarr, R. W., Cintron, L. M., Jiang, H. & Goldberg, B. (2013). Creating an Advanced Pedagogical Model to Improve Intelligent Tutoring Technologies. Paper presented at the Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

ABOUT THE AUTHORS

***Dr. Benjamin Goldberg** is an adaptive training scientist at the Army Research Laboratory's Human Research and Engineering Directorate. He leads research focused on instructional management within ARL's Learning in Intelligent Tutoring Environments (LITE) Lab and is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT).*

Developing a Pattern Recognition Structure to Tailor Mid-Lesson Feedback

Jeremiah T. Folsom-Kovarik[1], Michael W. Boyce[2]
[1]Soar Technology, Inc., [2]U.S. Army Research Laboratory

INTRODUCTION

The Generalized Intelligent Framework for Training (GIFT) has the potential to increase the micro-adaptive individualization of many training systems by overlaying adaptive feedback to learners during training sessions. For example, GIFT can augment a particular scenario in a first-person infantry simulation without needing to change the scenario itself, by displaying feedback messages in the tutor user interface (TUI) when particular learner experiences are observed. Feedback that GIFT delivers in the TUI can be as effective as feedback embedded directly in the system (Goldberg & Cannon-Bowers, 2015).

Expected observations (such as learner inputs or actions) that should trigger a response in GIFT are typically defined via a domain knowledge file (DKF). Importantly, feedback that responds to domain observations is best tailored to individual learners' needs when GIFT can select and deliver it on the basis of a rich collection of actionable information about learner experiences and characteristics. To this end, the DKF is enhanced with a new ontology of *patterns* that draw information from the relationships between single observations. Examples of patterns include order, timing, and repetition relations between observations.

A powerful existing tool to author and identify patterns is the Student Information Model for Intelligent Learning Environments (SIMILE) (Mall & Goldberg, 2014). The present work is compatible with SIMILE to the extent that SIMILE generates conditions which can be processed as input. Relative to SIMILE, the present research adds domain-general reasoning about features extracted from the domain-specific patterns. Furthermore, because it is native GIFT code, the present contribution is possible to use with GIFT Cloud. Interpreting patterns within GIFT's learner module and pedagogical module can increase their power to recognize and respond to proper performance in the training domain, learners' skill and knowledge, and inferences about learners' cognitive states and traits.

An initial demonstration of the work is being constructed for a military cognitive-perceptual training task that combines social and tactical challenges within each scenario. The demonstration uses patterns to define expected timing and order of responses in the domain, infer the latent mental processing steps of individual learners, and respond to learners with immediate formative feedback.

COGNITIVE-PERCEPTUAL TRAINING DOMAIN

Initial experimentation is grounded in a software system for tailored training and assessment previously created by SoarTech under DARPA funding (Hubal, van Lent, Marinier, Kawatsu, & Bechtel, 2015) known as Adaptive Perceptual and Cognitive Training System (APACTS). During the present research and development, APACTS is being modified to work with GIFT as an external training application and will be made available to GIFT users.

APACTS contains challenging, realistic decision-making scenarios developed in conjunction with experienced operators from Army and other training domains. The target training audience is an Army small-unit leader. The military battlespace where these leaders operate is characterized by uncertainty due to missing information, time pressure from the need to take advantage of tactical situations quickly, and high complexity with many interacting factors to consider (Thunholm, 2005). At the same time, the leaders' quick decisions can have far-ranging impacts on the larger U.S. mission (Malone, 1983).

APACTS scenarios test learners' decision-making ability in scenarios that draw on both tactical and social skill in the same scenario. APACTS sequences video, images, and two types of assessments: multiple-choice decisions and a perceptual task (Figure 1) that lets learners annotate images with specific visual cues that occur in the scene. Feedback is delivered via an after-action review (Figure 2). Key to the present work is that GIFT adds tailored mid-lesson feedback to APACTS via the TUI. GIFT selects feedback by recognizing and interpreting patterns in the learner performance during APACTS scenarios. The approach is general and may also be used to find and respond to patterns in other GIFT training tools.

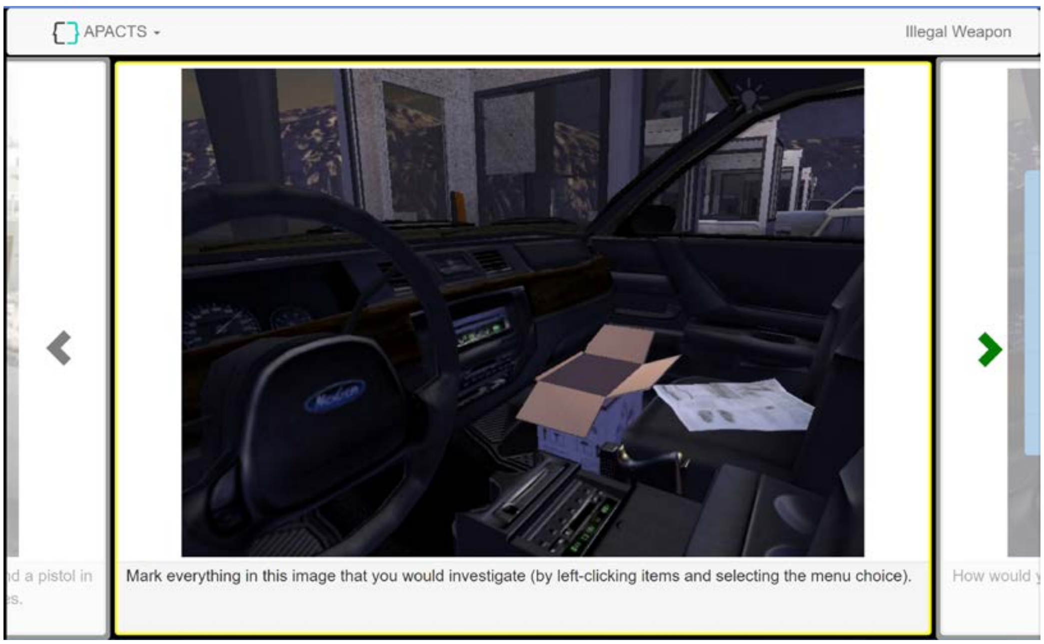


Figure 1. The APACTS cognitive-perceptual assessment tasks assess how learners process visual cues.

Terminology

For the purposes of this research, an *observation* is defined as a provable fact about what one learner has done within a learning tool. Examples of observations are *the learner opened a door* or *the learner scored 15 out of 20*. An observation happens at a single point in time, does not have duration, and has always either happened or not happened. An observation does not have a likelihood, does not need to be inferred, and cannot be incorrect. Any uncertainty surrounding an observation is assumed to be resolved by the training tool where the observation originated.

Within GIFT, individual observations are assembled into *patterns* via new additions to the domain module. Patterns are groups of observations that take on meaning in relation to each other. For example, clearing a room might require two observations within some period: *the learner opened a particular door* and then *the learner moved to the right*. The two observations might be related by ordering (one before the other) and by timing (one immediately after the other). Patterns may also be grouped and nested to arbitrary depth.

While occurrences of observations and patterns are both considered incontrovertible facts in the GIFT point of view, inference comes into play when the DKF defines *constraints* on observations and patterns. The satisfaction or violation of these constraints lets GIFT detect learner errors (constraint violations) and infer what *misconceptions* might underlie the observed performance. Misconceptions encode predictable but incorrect cognitive processing (Koedinger, Corbett, & Perfetti, 2012; Sleeman, Ward, Kelly, Martinak, & Moore, 1991). Misconceptions in GIFT extend the domain concept objects with new information that can help tailor feedback and provide appropriate pedagogical strategy.

During the present work, APACTS was instrumented with a typical interop plugin that communicates learner performance to GIFT through the gateway module and domain module. Because of these changes, GIFT has visibility into observations of individual learners as they progress through APACTS. This provides a testbed for demonstration and evaluation of the new GIFT capabilities to observe patterns in learner performance, infer errors and misconceptions, and tailor mid-lesson feedback.

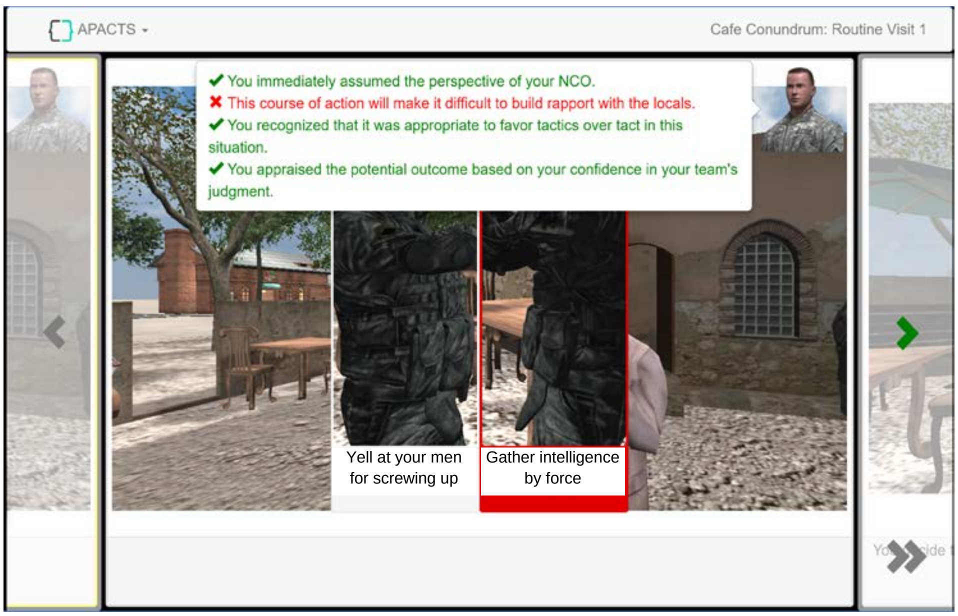


Figure 2. APACTS combines tactical / social decision-making assessments and AAR feedback.

OBSERVABLE PATTERNS

Patterns relate multiple observations to each other in time. Formal temporal logic is well studied in the context of, for example, characterizing software synchronization and timing (Clarke & Emerson, 1981) or reasoning about plans (McDermott, 1982). Patterns of observations that are implemented in GIFT represent a subset of temporal logic operators defined by Allen and Ferguson (1994). The patterns chosen for implementation reflect those hypothesized to be valuable for instructors or instructional designers to describe learner performance in a variety of modern training tools, to include APACTS and more sophisticated open-world simulations.

The GIFT team has made authoring tools a strong emphasis of the adaptive training program. Existing research has identified lessons learned for authoring tools in GIFT explaining the importance of empowering the user, which will build trust and confidence in the authoring process (Ososky, 2016). Authorability of patterns by nontechnical personnel is a key design consideration when choosing and defining patterns.

When future work adds the new patterns into GIFT authoring tools, it is vital that they align with how instructors think about learner performance. Instructional alignment may be more valuable than complete expressivity of the language when it can prevent errors, reduce cost of authoring, and increase technology acceptance among instructors and instructional designers (Folsom-Kovarik, Wray, & Hamel, 2013). For this reason, limiting the patterns that an author can express may actually improve the utility of the new constructs more than making it possible to express many more patterns but also requiring an engineering or mathematical background to get the patterns right. Furthermore, even the simplified syntax used in this paper may be hidden from nontechnical authors by presenting a graphical interface such as the draggable box and line diagrams described in (Woods, Stensrud, Wray, Haley, & Jones, 2015).

Although actions of other learners (or constructive characters) may cause observations and otherwise affect learners, the present work focuses on the individual learner use case. At present the GIFT patterns do not include a full definition of patterns that could be expected in a team environment.

Required, Forbidden, and Optional

The basic elements of patterns provide building blocks that instructors can assemble to describe learner performance in a training tool. The definitions of the basic elements in GIFT are based on constraint logic built for the Dynamic Tailoring System (DTS) (Wray & Woods, 2013).

First, observations may be *required*, *forbidden*, or *optional* (Table 1). These generalized basic ideas are already present in GIFT and implemented by individual domain module conditions. They help define types of errors that can let GIFT differentiate the cognitive processing in this learner, such as insufficient automaticity or presence of specific misconceptions, that led to the incorrect performance (Woods et al., 2015).

Table 1. The basic building blocks of patterns are single observations.

Notation	Meaning
A	*Required*. GIFT must observe condition A. The learner must carry out step A.
~A	*Forbidden*. GIFT must not observe condition A to be true. The learner must not do A.
A*	*Optional*. GIFT may or may not observe condition A; it is not required or forbidden.

In the absence of other constraints defined below, failure to observe a required element constitutes an *error of omission* that GIFT detects when the required element goes out of scope without being observed. The observation of a forbidden element constitutes an immediate *error of commission*. In the formal study of errors – for example in adverse event analysis (Donchin, Gopher, Olin, Badihi, Biesky et al., 1995) or in the human factors design of a system (Boyce, Sottilare, Goldberg, & Amburn, 2015) – errors of omission and errors of commission are typically considered to arise from different cognitive pathways and demand different diagnosis. An error of omission reflects a failure to carry out a required step, perhaps forgetting or not recognizing the need to do so. An error of commission is an incorrect action or actively doing what should not be done. Other error categories, such as sequence errors or context errors, are discussed below.

Optional elements can never cause a constraint violation, but when they are observed they can cause other processing in conjunction with the below patterns. Required and optional elements may be observed more than once without causing an error. To limit the repetitions allowed, see *repetitions* below.

Clusters, Dependency, and Strict Ordering

Clusters of observations are unordered sets that group elements together for checking. Clusters may have any number of members and may be nested to arbitrary depth. Importantly, checking a cluster can imply either a logical *AND* or a logical *OR* relation between members, depending on where the cluster appears. This helps move away from a strict temporal logic and toward a language of patterns that should match the intuitions of nontechnical users for teaching and training. Examples of the difference appear in Table 2: compare the logical processing implied by (A B) → C as contrasted with A → (B C).

A *dependency* relation between two elements (observations or groups of observations) indicates that the second element should not be observed before the first. For example, a learner on a patrol mission should not proceed outside the wire before completing a mission briefing. By contrast, a *strict ordering* relation indicates that not only must the second observation come after the first, but also the first observation becomes forbidden, and may no longer be repeated, after the second is observed. It is still permissible to observe A or B multiple times each, as long as they are not out of order.

Both dependency relations and strict ordering relations may be chained to arbitrary length, as shown in Table 2. When several elements participate in a strict ordering, there is no way to exempt a member element from being strictly ordered (no partial ordering).

Table 2. Clustering and ordering multiple observations.

Notation	Meaning
(A B)	*Cluster*. A and B are separate observations, but are checked together. They are required.
~(A B)	Both A and B are forbidden.
(A B)*	Both A and B are optional.
A → B	*Dependency*. B depends on A. The learner cannot do step B without first doing step A.
A → B*	B does not need to be observed, but if it is observed then it triggers an error of omission unless A is observed first.
A* → B	Equivalent to just B, because A can either happen or not happen before B is observed. Note that this is a change from interpretation in the SoarTech DTS.
(A B) → C	Both A *and* B must be observed before C may be observed.
A → (B C)	A must be observed before either of B *or* C may be observed.
(A, B)	*Strict order.* A and B must be observed in order. It is not allowed to do B until A is done, and also it is not allowed to do A after B.

Note that the strict ordering (A, B) is simply equivalent to A → (B ~A). The definition of strict ordering as another first-order constraint becomes useful for authors when there are several elements that need to be ordered. For example, (A, B, C, D) is easier to encode than A → (B ~A) → (C ~A ~B) → (D ~A ~B ~C). Furthermore, if a graphical UI is used to author these relations, fewer nodes and edges will be required.

Relevance and Exclusivity

Relevance refers to the notion that required or forbidden elements (usually clusters) will not be checked under certain conditions when they are not instructionally relevant. Relevance is similar to the concept of scope in computer programming. For example, checking whether a learner clears a room correctly is (at first) not relevant in the context of a patrol scenario, but may become relevant if the patrol comes into contact with the enemy and must conduct a tactical engagement.

Controlling the relevance of an element is not simply a matter of saving computational resources. It is also important instructionally. Relevance can be used to make assessment tractable in ill-defined domains (Nye, Boyce, & Sottilare, 2016; Woods et al., 2015). For example, if a learner on patrol enters a village elder's home for tea, it would be inappropriate for GIFT to state he made an error by not clearing the room first.

Exclusivity refers to the requirement that all unmentioned observations are considered forbidden. Exclusivity can be specified at the same level of granularity as any cluster, including a top-level luster that contains all others. Otherwise, if a cluster is not marked as exclusive, any unmentioned observations are considered optional. When a cluster is not relevant, its exclusivity constraint is not checked.

Repetition, Pause, and Duration

Elements may be *repeated* a number of times that instructors specify. For example, in a patrol scenario the learner might need to greet between two and four civilians in the local language. Any number of occurrences in the range satisfies the constraint. An exact value can also be specified. The repetition constraint does not rule out other observations between the repetitions or after them.

A *pause* is an interval between two observations. The time starts counting every time the left-hand side of the constraint is satisfied. If the right-hand side becomes satisfied before the minimum specified delay, then the delay constraint is violated. If the maximum specified delay expires and the right-hand side is not satisfied, the delay constraint is violated.

Finally, *duration* describes how long it should take the learner to complete one or more observations. The entire cluster must be satisfied within the timespan specified by max. If the time specified by max elapses and the cluster is not satisfied, the constraint is violated. Like all constraints, durations may be nested, enabling a series of observations that have a total time for completion and duration for each individual item. This method is valuable when studying speed-accuracy tradeoffs (Goldhammer, 2015).

Table 3. Patterns of repeating and timed observations.

Notation	Meaning
A r[min..max]	*Repetition.* Element A must be observed at least *min* times and at most *max* times.
A r[exact]	Element A must be observed exactly *exact* times.
A r[2]	The learner must do A twice. If the learner does A three times, no error happens.
A r[2] → B	The learner must do A twice before doing B.
A r[2] → ~A	The learner must do A twice, after which the learner may not do A again.
→ p[min..max]	*Pause.* The time between these two observations must be between *min* and *max*.
A → p[30 sec ..] B	B must occur after A and also at least 30 seconds must separate them.
A → p[.. 30 sec] B	B must occur after A and also within 30 seconds after A is observed.
A d[max]	*Duration.* Element A will be relevant for up to *max* seconds.
(A B) d[30 sec]	The learner has 30 seconds to complete A and B.
(~A) d[30 sec]	The learner may not do A for the first 30 seconds that the constraint is relevant.

In the present work, patterns that express order, repetition, and timing form the basis for infering general insights about learners and improving feedback tailoring.

INTERPRETING PATTERNS AND TAILORING FEEDBACK

The first modification in GIFT to take advantage of information from observable patterns is the idea of a *misconception*. Misconceptions modify concepts within the domain module. They give GIFT additional information about learner performance – not just whether a concept has been mastered or not, but also inferences about why a concept may not be mastered and what specific feedback may be needed.

Misconceptions have been well studied elsewhere and evidence exists that they are valuable to providing tailored feedback. A few example benefits are listed. Detecting and addressing specific misconceptions can challenge learners' incorrect mental models when untailored feedback would otherwise allow them to gloss over the differences (Swan, 1983). Feedback focusing on misconceptions is also more directive, when GIFT detects that such feedback is more appropriate for an early stage of learning (Moreno, 2004) than an alternative facilitative feedback or an exploration experience during later stages. Inferring the presence of misconceptions can also support increased specificity in feedback which is appropriate when learners are more performance oriented (Davis, Carson, Ammeter, & Treadway, 2005). In conjunction with GIFT's *active* and *constructive* feedback mechanisms, the addition of misconceptions will help to provide feedback that aligns with many guidelines for delivering formative feedback (Shute, 2008).

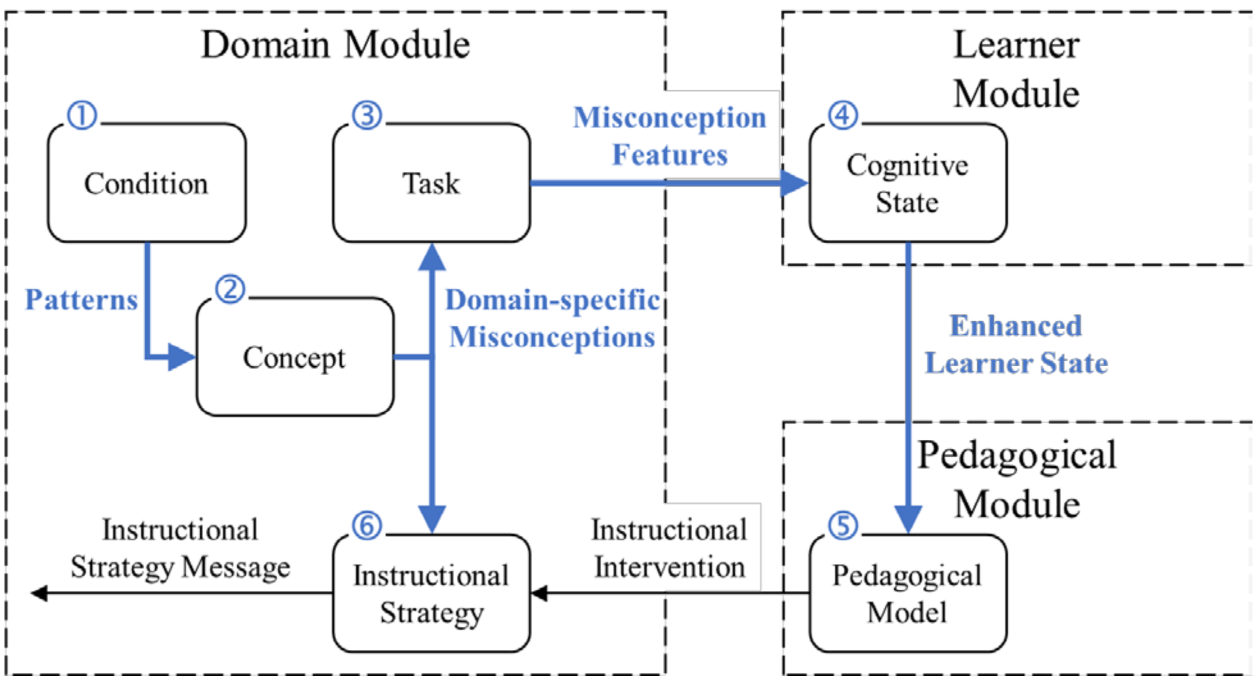


Figure 3. High-level data flow for inferring misconceptions and using them to tailor feedback.

Figure 3 depicts a high-level data flow for observing patterns of learner behavior and inferring the presence of misconceptions. Selected relevant classes within each GIFT module are shown with blue (bolded) lines indicating changes to the standard GIFT classes and messages.

First, GIFT patterns implement new kinds of conditions (1). Like conditions, the patterns can modify the state of domain-specific concepts (2). Concepts are similarly modified to contain an arbitrary number of misconceptions, each of which is tied to its parent concept. So, it becomes possible to differentiate between specific ways that a learner may act or know incorrectly. The different misconceptions may require different levels of urgency or different modes of feedback.

While misconceptions within the domain module are domain-specific, GIFT needs to reason about misconceptions in a domain-general manner within the learner module and pedagogical module. For this reason, dimension reduction in the domain module passes along only a subset of features for each detected misconception. The extracted misconception features are domain-general and include the importance, urgency, and certainty of each misconception. Determination of these values is task-specific (3). For example, in a VBS scenario the domain module might detect a pattern of learners walking around with weapon in the wrong ready state. In a squad tactics setting this might be an unimportant error, while the error would be more important if the scenario is targeting intercultural communication with civilians while on a patrol.

Once the learner module has domain-general information about misconceptions as they are detected, stored in the cognitive state class (4), the pedagogical module gains new information within the learner state message set on which to base real-time tailoring decisions. While performance assessment messages are domain-specific, the misconception features are generalized and thus update the learner cognitive state. Within the scope of the present research, a simple algorithm will be added to an existing pedagogical model (5) that acts on the domain-general features of misconceptions to direct instructional interventions. For example, the initial pedagogical algorithm might indicate that some number of unimportant misconceptions may be addressed through AAR or reteaching, while any misconception with importance above some threshold must be addressed through immediate feedback.

Finally, when the pedagogical module requests an instructional intervention, the domain module contains the full misconception information that is required to deliver needed feedback with high specificity (6).

APACTS Examples

Two examples suggest the value of leveraging observable patterns in GIFT.

First, Figure 1 above depicts an example of a visual scan task in APACTS. The learner is stationed at an entry control point and must respond to a civilian vehicle as shown in a static image. Optionally, there is a time limit on the learner’s response. The correct response is to mark two objects in the Figure 1 image: the box on the passenger side floorboard, and a pistol grip that is visible between the center console and the passenger seat (Figure 4). However, by using a new observation ordering pattern, GIFT can now add specificity to the APACTS assessment of correct behavior.

Figure 4. Detail of Fig 1, highlighting threat item.

This image is designed so that the more threatening object, the pistol, is less visually salient (less noticeable) compared to the box, which is easier to see because it is larger and a lighter color. Since APACTS communicates each click the learner makes to GIFT, it is easy for GIFT to define domain-specific constraints that not only require clicking on both objects, but also differentiate between which object was clicked on first. If the learner clicks the box before the gun, that ordering may be caused by a more reactive cognitive processing of the scene (Schatz, Colombo, Dolletski-Lazar, Carrizales, & Taylor, 2011), and can be associated with the inference this is a more novice learner. If the learner clicks the gun first, that observation provides evidence that the learner is more expert in visual scene assessment.

As a second example, Figure 2 above depicts a typical multiple-choice assessment in APACTS (although overlaid with the built-in AAR feedback). Multiple-choice assessments provide an opportunity to gather information via observation timing patterns. With these, GIFT can make use of observations that a human instructor might value such as the amount of time the learner considered the question before making a choice. Fast choices might be associated with a more expert learner. GIFT can also make use of information such as whether the learner changed between choices before submitting, or simply hovered the mouse over one option or the other, to differentiate hesitation from other reasons for delay such as inattention.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

In conclusion, GIFT is being enhanced with new domain-specific and domain-general representations of learner performance and underlying cognitive state that will make tailored feedback specific and impactful.

The implementation status of the work described includes initial changes to the GIFT source code in a development branch. The changes will be made available to the GIFT community in the future, after appropriate code review. APACTS software and scenarios will be published and made available.

A demonstration is planned via a human-participants study of APACTS. The demonstration is expected to compare training efficacy using new, tailored feedback against the baseline of APACTS alone. In addition, the implementation work supporting the study may be reused as a publicly available reference or showcase of the new capabilities and how to use them.

Future development work will include adding the new patterns into GIFT authoring tools. Finally, the patterns will be demonstrated on a second domain besides APACTS. That work will demonstrate the generality of the approach and utility to enhance widely used tools such as VBS or other training systems.

Finally, interesting directions for future funded research might include machine learning of patterns such as time limits that differentiate different cognitive processing pathways, helping to assess automaticity of skill performance. GIFT research efforts such as metacognition assessment, or active and constructive interventions, should also be combined with this work in order to improve the simple tailoring algorithms.

ACKNOWLEDGEMENT

This material is based upon work supported by the Human Research & Engineering Directorate - Advanced Training and Simulation Division (HRED - ATSD) and the Army Contracting Command (ACC) - Aberdeen Proving Ground (APG) - Research Triangle Park Division (ACC-APG-RTP), Orlando Branch under Contract No. W911NF-16-C-0063**.** Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ATSD.

REFERENCES

Allen, J.F., & Ferguson, G. (1994). Actions and events in interval temporal logic. *Journal of logic and computation, 4*(5), 531-579.

Boyce, M.W., Sottilare, R.A., Goldberg, B., & Amburn, C.R. (2015). *Work Domain Analysis for Ecological Interface Design of Tangible Interfaces*. Paper presented at the Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), Orlando, FL.

Clarke, E.M., & Emerson, E.A. (1981). *Design and synthesis of synchronization skeletons using branching time temporal logic.* Paper presented at the Workshop on Logic of Programs.

Davis, W.D., Carson, C.M., Ammeter, A.P., & Treadway, D.C. (2005). The interactive effects of goal orientation and feedback specificity on task performance. *Human Performance, 18*(4), 409-426.

Donchin, Y., Gopher, D., Olin, M., Badihi, Y., Biesky, M.R., Sprung, C.L., et al. (1995). A look into the nature and causes of human errors in the intensive care unit. *Critical care medicine, 23*(2), 294-300.

Folsom-Kovarik, J.T., Wray, R.E., & Hamel, L. (2013, July 9-13). *Adaptive assessment in an instructor-mediated system.* Paper presented at the 16th International Conference on Artificial Intelligence in Education, Memphis, TN.

Goldberg, B., & Cannon-Bowers, J. (2015). Feedback source modality effects on training outcomes in a serious game: Pedagogical agents make a difference. *Computers in Human Behavior, 52*, 1-11.

Goldhammer, F. (2015). Measuring ability, speed, or both? Challenges, psychometric solutions, and what can be gained from experimental control. *Measurement: interdisciplinary research and perspectives, 13*(3-4), 133-164.

Hubal, R., van Lent, M., Marinier, B., Kawatsu, C., & Bechtel, B. (2015). *Enhancing Good Stranger Skills: A Method and Study*. Paper presented at the Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), Orlando, FL.

Koedinger, K.R., Corbett, A.T., & Perfetti, C. (2012). The Knowledge-Learning-Instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive science, 36*(5), 757-798.

Mall, H., & Goldberg, B. (2014). *SIMILE: an authoring and reasoning system for GIFT.* Paper presented at the 2nd Annual GIFT Users Symposium.

Malone, D.M. (1983). *Small unit leadership: A commonsense approach*: Random House Digital, Inc.

McDermott, D. (1982). A temporal logic for reasoning about processes and plans. *Cognitive science, 6*(2), 101-155.

Moreno, R. (2004). Decreasing cognitive load for novice students: Effects of explanatory versus corrective feedback in discovery-based multimedia. *Instructional science, 32*(1-2), 99-113.

Nye, B.D., Boyce, M.W., & Sottilare, R.A. (2016). Defining the Ill-Defined: From Abstract Principles to Applied Pedagogy *Design Recommendations for Intelligent Tutoring Systems: Volume 4-Domain Modeling*.

Ososky, S. (2016). *Practical requirements for ITS authoring tools from a user experience perspective.* Paper presented at the International Conference on Augmented Cognition.

Schatz, S., Colombo, G., Dolletski-Lazar, R., Carrizales, M., & Taylor, A.H. (2011). *A Proposed Taxonomy for Perceptual—Cognitive Skills* Orlando, FL: MESH, a DSCI Company.

Shute, V.J. (2008). Focus on formative feedback. *Review of educational research, 78*(1), 153-189.

Sleeman, D., Ward, R.D., Kelly, E., Martinak, R., & Moore, J. (1991). An Overview of Recent Studies with PIXIE. In P. Goodyear (Ed.), *Teaching knowledge and intelligent tutoring* (pp. 173–186). Norwood, NJ: Ablex Pub. Corp.

Swan, M. (1983). *Teaching decimal place value: A comparative study of "conflict" and "positive only" approaches*: Nottingham University Shell Centre for Mathematical Education.

Thunholm, P. (2005). Planning under time pressure: An attempt toward a prescriptive model of military tactical decision making *How professionals make decisions* (pp. 43-56).

Woods, A., Stensrud, B., Wray, R., Haley, J., & Jones, R. (2015). *A Constraint-Based Expert Modeling Approach for Ill-Defined Tutoring Domains.* Paper presented at the Twenty-Eighth International FLAIRS Conference.

Wray, R.E., & Woods, A. (2013). *A Cognitive Systems Approach to Tailoring Learner Practice.* Paper presented at the 2nd Advances in Cognitive Systems Conference, Baltimore, MD.

ABOUT THE AUTHORS

Dr. Jeremiah T. Folsom-Kovarik *is a lead scientist with Soar Technology, Inc. He earned a Ph.D. in Computer Science from the University of Central Florida in 2012. Dr. Folsom-Kovarik's research focuses on novel methods, algorithms, and knowledge representations that help automate (1) capturing and assessing sensor inputs with an emphasis on monitoring trainee performance, (2) modeling underlying reasons for observations and inferring individual trainee needs, and (3) automating intelligent systems that can reason about the contexts, goals, and tradeoffs required to carry out effective, individualized skill training.*

Dr. Michael W. Boyce *is a research psychologist at the Army Research Laboratory, Human Research and Engineering Directorate (ARL-HRED) supporting the Adaptive Training Research Program and the Advanced Modeling and Simulation Branch. His research focuses on the integration between the Generalized Intelligent Framework for Tutoring (GIFT) and other training applications such as the Augmented REality Sandtable (ARES). He earned his doctorate in Applied / Experimental Human Factors Psychology from the University of Central Florida in 2014.*

Integrating the outer loop: Validated tutors for portable courses and competencies

Elliot Robson[1], Fritz Ray[1], Anne M. Sinatra[2]
[1]Eduworks Corporation, [2]U.S. Army Research Laboratory

INTRODUCTION

The Generalized Intelligent Framework for Tutoring (GIFT) is a broad and flexible framework for developing and delivering training. GIFT currently provides support for best practices and most effective pedagogy in the tutoring context, however, it is limited in providing support in the course development process. In this paper, we present the GIFT Structural Equation Modeling (GIFT-SEM) module for doing rapid analysis and validation of the concepts and concept models used to guide training development. We believe this is one piece of a suite of validation tools that will be used to allow GIFT authors to identify and share effective pedagogy, concept models, sequencing, and learning resources. This validation suite will provide the evaluation necessary to integrate GIFT tutors into synthetic training environments (Dumanoir, 2015) and larger service oriented architecture (SOA) ecosystems. Additionally, validated tutors and pedagogy allow authors to learn from each other's work, share validated training throughout the GIFT ecosystem, leverage near and far transfer of learning, and validate external resources within their domain.

Validation is not only central to a vision of GIFT performing a complete iterative instructional design loop from conceptualization to evaluation, but it supports periodic refinement of instructional materials, which is important for extending the lifespan and reuse of training. Most successful long-term training development and creation strategies include validation and iterative improvement. For this discussion, we chose to use the ADDIE process (Figure 1) because it is a general process used by many instructional designers (Molenda, 2003). ADDIE: analyze, design, develop, implement, and evaluate, is an iterative process that constructs and refines instructional material.

In GIFT, the ADDIE process has been supported through the use of tools to assist in each stage of the process. Analysis of the subject domain can be done by subject matter experts, or through tools such as TRADEM (Ray, 2014). Design, Development, and Implementation of the tutor are accomplished through GIFT Authoring Tools, Content Authoring, GIFT Cloud, and more. With the integration of the SEM component, GIFT has the first piece of the evaluation module necessary to complete the ADDIE process (Branch, 2009).

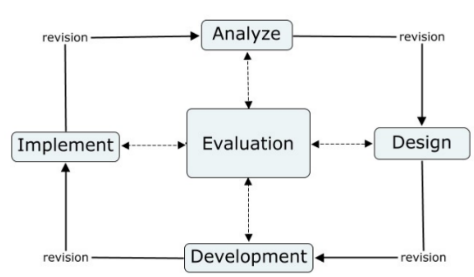


Figure 1: ADDIE Process

In this paper we will discuss authoring in GIFT and how GIFT-SEM provides a roadmap for integrating evaluation into the GIFT Cloud workflow. In addition, we will outline a broader GIFT evaluation module along with a description of how integrating GIFT into a SOA can extend the ADDIE process and enhance the capabilities of GIFT. Finally, we will discuss the long-term advantages of a full instructional design loop in GIFT and how GIFT can inform a larger SOA.

BACKGROUND

Course Authoring In Gift

GIFT is a domain-independent framework for creating intelligent tutoring systems (ITSs). GIFT has been designed to promote reusability of materials, reduce the time that it takes to author an ITS, and lower the skill level needed to author an ITS (Sottilare, Brawner, Goldberg, & Holden, 2012). Due to the generalized nature of GIFT, there is a great deal of flexibility available in regard to how to author a course. Based on the subject matter, the student level, and even the preference of the course designer there may be different functionality that is utilized on a course-to-course basis.

The GIFT Authoring Tool has been updated and refined over time, and allows the individual who is creating the course to select and organize the course elements that they will need, the ability to create surveys/question banks, and the ability to set up an adaptive courseflow. Before beginning the authoring process, the concepts that are associated with the course should be determined and established by the author based on their own instructional design and domain knowledge. These concepts are then used as tags throughout the course authoring process. The GIFT course authoring process supports most common forms of content, including: word, pdf, ppt, HTML, slideshows, and more. A screenshot of the GIFT Cloud Authoring Tool interface as of GIFT 2017-1 is in Figure 2.

In the ideal GIFT authoring process, the individual who was authoring the course would bring all of their course materials, survey questions, and desired adaptations and work with the system to create their adaptive course. The design and ordering of the material that they use would be up to them, but may be informed by best practices, external concept models or the user's knowledge/beliefs about the content. In addition, the GIFT-SEM module allows users to validate these beliefs about course sequencing. When implemented, knowledge extracted from courses using the GIFT-SEM evaluation module can also be used as a basis for the selection of course order. The ability to use GIFT evaluation modules to assess the current state of a course based on learning outcomes allows GIFT to provide a clear path to iterative refinement of the tutor, eventually resulting in a proven and validated tutor.

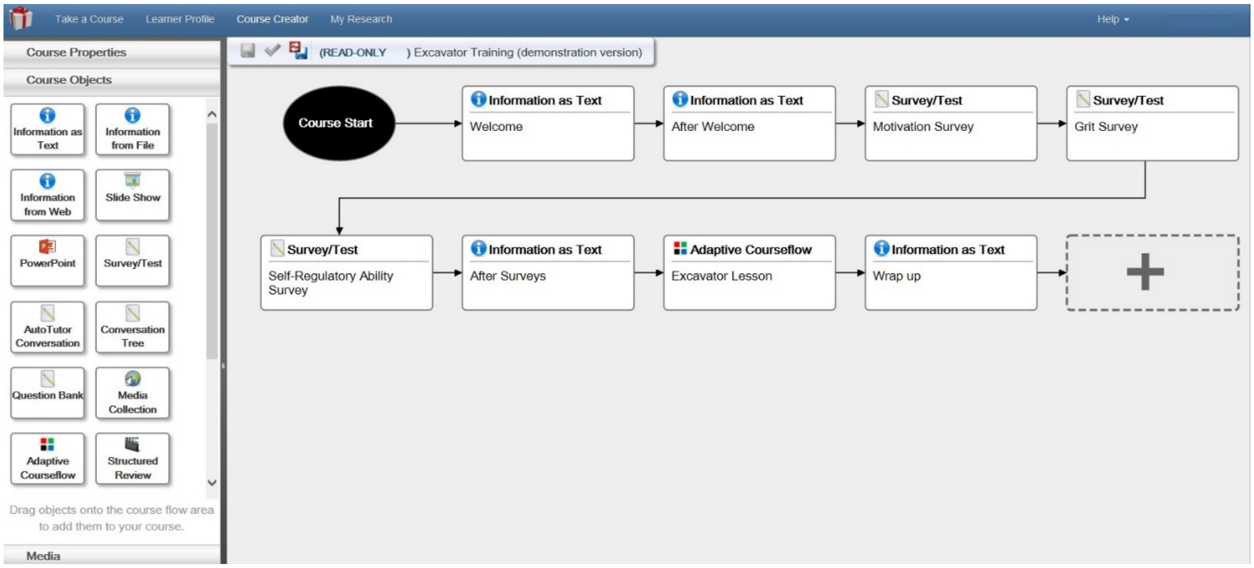


Figure 2: GIFT Authoring Tool Interface. The Course objects are on the left. The course flow is on the right.

GIFT-SEM in GIFT

GIFT-SEM is a tool that performs analysis and evaluation of a tutor, and provides a report describing the general 'fit' of learner data with the course sequence and concepts. To use GIFT-SEM, an author creates a course, utilizing concepts, surveys and question banks, optionally skipping content. During the course creation process, the user translates their beliefs about the ordering of underlying concepts into pedagogy through the sequencing of elements and the concepts that elements are tagged with. This concept map, as represented by the concepts selected during course creation, is the main target of analysis for GIFT-SEM.

After the course has run, the author uses GIFT-SEM to analyze the course based on the performance of students during the course. Finally, the author then publishes and collects results in an experiment and uses the GIFT-SEM tool to perform analysis. This analysis produces a report (Figure 3) that provides a high-level summary of how well the author's concept model is supported by the student results. In addition, the final output details the statistical checks used to reach this conclusion. This allows users with multiple levels of statistical sophistication to use GIFT-SEM effectively. These reports allow all users to identify cases where their beliefs about the concepts and concept ordering of a course is not supported by actual student results. In cases where the results are poorly supported, it is possible to use GIFT-SEM to explore other hypothetical concept mappings. This "what if" exploration allows users to discover more effective pedagogic sequencing and identify their own misconceptions about the concepts that underlay content.

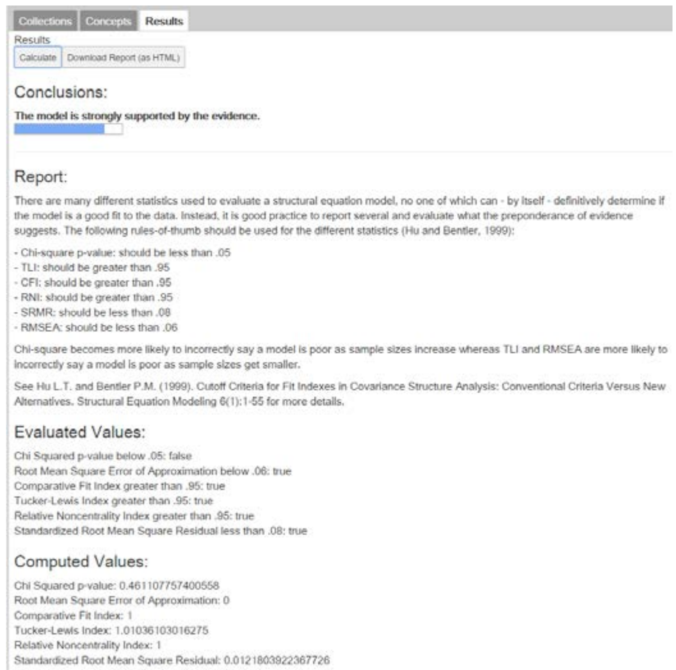


Figure 3: GIFT-SEM Detailed Report

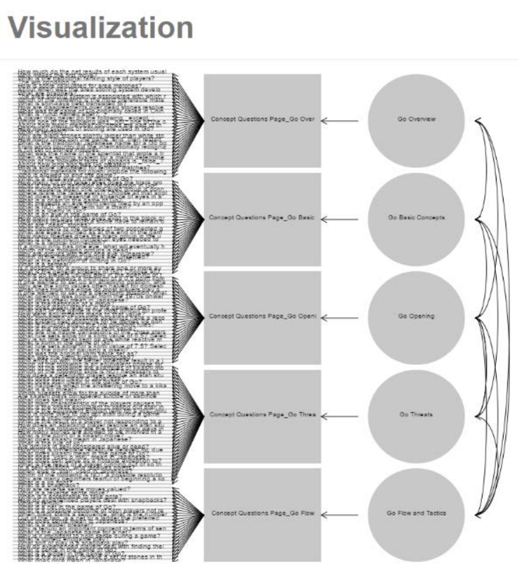


Figure 4: GIFT-SEM Concept Map Visualization

GIFT-SEM is the first internal evaluation tool for GIFT. It allows experts and authors to evaluate and validate concept models used to design and develop training using SEM. Using actual student results, GIFT-SEM analyzes how well the learning outcomes support the concept model and suggests easily understood next steps to the user. GIFT-SEM supports two primary types of investigation: confirmatory analysis where an expert uses SEM results to confirm or reject the concept model; exploratory analysis where an expert tests different concept models against student outcomes to see how well a candidate model is supported by the evidence.

In order to successfully apply causal modeling (SEM), it is necessary to identify the concept model being used by a piece of training or content. Currently, GIFT-SEM extracts the concept model by tracing the learning path using the concepts mapped in GIFT to assessments and other content. GIFT Cloud provides data through xAPI and GIFT's internal logging format and tools. However, extracting the information necessary to identify the learner's path and support causal analysis requires an additional interpretation layer to be applied to GIFT logs. The current interpretation layer built for GIFT-SEM is designed with a larger GIFT evaluation module in mind: it extends the existing GIFT data model and provides support for the evaluation of pedagogic approaches, training outcomes, and content ordering. In GIFT-SEM, we visualize the concept map portion of the interpretation layer's analysis of the GIFT logs (Figure 4). The GIFT-SEM visualization gives a high-level view of the gift course from a concept map perspective. In Figure 4, the left side shows every question contained in the course, with the surveys in the middle and finally the concepts on the right. Links between concepts are inferred from the logs based on author tagging in the GIFT authoring tool. We believe that the interpretation layer used in GIFT-SEM to analyze and visualize GIFT log data could be reused and expanded in other analysis applications and support non-xAPI data sharing between GIFT and other software.

SOAs in Education

Validation of tutors is important to GIFT as a method of improving pedagogy, but it is also a core function needed to cement GIFT as the central authority within larger systems that make use of multiple pieces of software. Specifically, SOAs are large, distributed collections of software components that fulfill the individual roles necessary to deliver a complete user experience (Papazoglou, 2003). In a SOA, some parts of the architecture are shared by all the components; for example, user management and student learning plans. By separating the role of each component, and defining clear protocols and boundaries, specific components can be reused across user experiences. This creates an ecosystem where there are many common components that provide large amounts of functionality, allowing features such as single-sign-on, shared learner profiles, shared concept models, common resource stores, and more.

SOAs, however, are relatively new to the domain of education (Lavendelis, 2012). While an educational SOA can utilize several common components, such as user management, content repositories, and more, there is less consensus among educational protocols for representations of learner knowledge, affective state, pedagogical practice, and other topics specific to education. Additionally, existing non-SOA Learning Management Systems (LMSes) provide a comprehensive, if monolithic, solution and remain difficult to displace.

Globally, SOAs are increasingly common and provide options for public and shared components common to the entire world (Bauer, 2013). In education, this opens up the potential to implement shared components from a broad number of sources, and focus development on specific educational problems and goals. In GIFT, integration into SOAs is an opportunity to integrate with other components that provide services beyond the GIFT framework. Evaluation modules in GIFT are a cornerstone of this effort because they change GIFT from an authoring tool within the SOA ecosystem into a provider of validated tutors that provide gold-standard information for other components.

GIFT IN AN SOA

GIFT Cloud, when embedded in one or more SOAs as shown in Figure 5, would enable use of the full ADDIE process to produce effective and evaluated tutors that interoperate with other components in the respective SOA to contribute to a persistent learner profile that syncs with the GIFT outer loop. This alignment between the GIFT learner data and the learner profile maintained by the SOA supports heavy customization of content both through adaptive course flows within GIFT and adaptive content from external components. With GIFT-SEM and other evaluation modules,

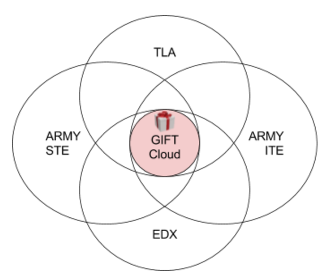


Figure 5: GIFT in SOAs

validated GIFT tutors will be able to interoperate among a network of validated learning activities, including smart content, ITSs, and simulations, providing a comprehensive, connected learning experience. This outcome is essential to systems such as the Army Integrated Training Environment (ITE) and future Synthetic Training Environments (USAAC, 2017). To support these use cases, we demonstrate the current GIFT Cloud workflow enhanced with new ADDIE-aligned processes that support shared competencies, metadata and other forms of interoperability.

Enhancing ADDIE for use in an SOA

GIFT is in the process of being integrated into a number of SOAs ranging from a SOA with persistent learner profiles, competency tracking, and content repository, the Total Learning Architecture (ADL, 2017), to coarse grained SOAs such as edX. To support an instructional design process that fits into a service oriented architecture that has components beyond GIFT, it is necessary to add new information at each step of the ADDIE process. The additional information needed to integrate ADDIE, as implemented in GIFT, into an SOA is outlined here:

Analyze: In the first step of the ADDIE process, a designer analyzes the domain of instruction with the assistance of a subject matter expert (SME). To maximize interoperability and compatibility, a list of concepts is constructed from existing knowledge of the domain. In well-articulated domains, such as mathematics or history, this is simple. In the absence of existing taxonomies, steps should be taken to use standard and reusable vocabulary in naming concepts, and the resultant list of concepts should be published and made ready for reuse. For example, in edX, an SOA, there are no authoritative concept lists, so authors must identify GIFT concepts for their course, and align them with other course learning objectives.

Design: The designer identifies, acquires, and conceives new content. This content is selected from other portions of the SOA that are focused on content creation, curation, storage, and alignment. In an integrated SOA, the GIFT evaluation/interpretation layer will align with skills, learning outcomes, competencies, and other learning objects used in the SOA. This content should be aligned at the appropriate level of granularity and maintain interoperability with the other SOA components. This is currently done by using standard forms like LRMI alignment syntax (ASSESSES, TEACHES, REQUIRES). One example SOA, the TLA, uses persistent universal resource identifiers (URIs) to maintain alignment between GIFT, content sources, competencies, and learner data.

Develop: The designer imports content and constructs the GIFT tutor, in accordance with the analysis and design. In this step, the designer uses pre-tests, adaptive tutoring elements, and other concept-aligned entries in GIFT, as well as ensuring correct reporting data is available to the GIFT Evaluation/Interpretation layer and other SOA components. The reporting data in the development step is crucial for validating the tutor, the design of the course, resources used in the tutor, and the concept model the tutor is based on. During this development phase, the metadata and information to be gathered from the tutor is selected. This reporting data includes any data the author or GIFT will to collect during the course. Static information and metadata is stored using persistent syntax like the LRMI alignment syntax, and activity data is generated by instrumenting the content with xAPI events. Currently, many SOA elements are designed so that they have a base set of xAPI statements that are always instrumented. For example, quiz questions will always report the answer entered by a student. Cutting edge software also includes ways for instructors to easily instrument additional xAPI events of interest for their course. For example, an instructor might want to know how long a student spent on a question. This reporting is a key feature to enable the course to adapt to the needs of the learner. By reporting data to the SOA, GIFT enables micro and macro-adaptivity.

Implement: The designer implements the tutor. Values for concepts used in GIFT may be acquired from local learner data, but may also be loaded from other portions of the SOA. Data and logs of student actions and results are stored locally while xAPI and other reporting protocols are used to inform other learning systems. For example, in the Army ITE vision, information about student learning and mastery can be passed between GIFT enabled training stations in a training center.

Evaluate: Finally, the designer may evaluate the course. Currently, GIFT-SEM is the only evaluation tool available in GIFT, so the only analysis provided by GIFT to the SOA is validation of the concept model. Validated concept models, and other future validated content, will allow GIFT to quickly repurpose, update, and reuse successful portions of content, as is normally done in ADDIE. It will also allow the evaluator to evaluate data in the context of the entire SOA, informing both the effectiveness of GIFT tutors and the quality of data and resources coming from other parts of the SOA.

EVALUATION

GIFT Evaluation/Interpretation Layer

The GIFT Evaluation/Interpretation Layer is currently built into the GIFT-SEM module to interpret GIFT logs and provide data and visualization of the concept map. However, this module has been designed to manage data extraction, interpretation, and formatting for use in multiple evaluation modules. We believe that this, or another interpretation layer like it, is key for centralizing GIFT as the heart of an SOA. The data may be ingested from two primary sources: GIFT logs and xAPI statements from GIFT or an SOA that GIFT Cloud is embedded in. By providing a central data store and format, the interpretation layer ensures that GIFT evaluation modules will maintain interoperability and the validations they provide can be transported throughout the GIFT and SOA ecosystem.

GIFT Cloud: Log Analysis

As a computerized tutoring system, GIFT produces vast quantities of information during execution that can be difficult to process without specialized tools and skilled personnel versed in analysis techniques. The implementation of GIFT-SEM focuses on providing a holistic analysis of the tutor as well as a guide to interpreting the output of that analysis. To accomplish this, GIFT-SEM parses logfiles generated in GIFT for events pertaining to particular concepts, such as the execution of a survey or question bank, the concept model's internal ordering of concepts, as well as learner, time and state data. Using this data, GIFT-SEM builds an intermediate model based on the concepts in GIFT with pathways informed by learner paths in the system. Data is attached to each question, survey, and concept node, compiled into a SEM model, and calculated. The process allows for modification of the intermediate model before SEM analysis is performed, which enables exploratory, and possibly other, forms of analysis.

SOA Integration

In implementations of SOA, there is an expectation that GIFT and other components shall provide reporting capabilities that affect both the real time operation of the system, and provide long term analysis and evaluation capabilities at a whole-system level. It is conceivable that in order to evaluate a GIFT tutor, one must consider information from other systems, especially if frameworks, models and other learning objects are shared between GIFT and other components. We foresee a bidirectional interaction with reporting technologies such as xAPI as well as common learner models to answer necessary questions of evaluation, to determine the effectiveness of a tutor in influencing future learning activities, and in incorporating information from real world application of skills taught by GIFT.

Validation and Transportability of GIFT Tutors

The above outlines tools needed for iterative refinement of tutors and sharing of validated best practices in GIFT and within an SOA ecosystem. Today, the use of GIFT-SEM allows for the user to explore multiple sequences and analyze how well different course sequences support the author's underlying concept model for the domain. With the addition of the tools discussed above, it is possible for GIFT users to analyze tutors in multiple ways (e.g., sequencing, pedagogy, competence) and make updates to the tutors based on these analysis. The process of results-driven updating not only creates validated tutors that provide more effective pedagogy, but is also a vehicle for higher-level transfer of best practices between authors. Using GIFT-SEM to validate a concept model for a domain has implications for the sequencing of that specific tutor, but also leads to a better understanding for that author of how the domain works. This kind of learning, when curated, is a valuable resource for new and experienced instructional designers. The creation of more evaluation modules adds additional dimensions to GIFT's ability to validate individual courses and identify best practices and pedagogy.

Today, the state-of-the-art in instructional design and content reaches beyond the boundaries of GIFT and into the SOA ecosystem to incorporate conceptual objects that represent and measure course and student data. Increasingly, a student profile that shares measurements of micro-skills, like how a soldier pulls the trigger on a rifle, and underlying macro-abilities, like mathematical intuition, is at the core of SOAs. For example, the competency frameworks used by the TLA allow for results from validated GIFT tutors to be flowed into other portions of the SOA, helping produce adaptive experiences that result in improved learner outcomes. The Army has comprehensive skill frameworks, found in Army Instructional Manuals that can be used to track readiness, adapt content, and select future programs of study. When a tutor is validated, its results are reliable, making them usable by other systems. When a tutor is validated, and GIFT is in an SOA ecosystem, its results are both reliable and portable to the rest of the components of the SOA. This is one way how advanced pedagogy, adaptive content and course-flow, and better student learning can be best supported.

There are three primary types of validation in GIFT that support the ADDIE process: validated concept models, validated pedagogy, and validated resources. Currently, GIFT-SEM allows for concept models to be validated and shared. The addition of more GIFT modules in the evaluation portion of ADDIE will support validation of pedagogy, sequencing, and resources resulting in fully validated GIFT tutors. When integrated into an SOA, this will allow for both transfer of effective best practices across authors and support new authors in developing higher quality tutors.

The validation of GIFT tutors in their respective domains is independently useful, however, alignment of tutors with learner profile data creates the possibility of adaptive content that adapts based on existing information known about the learner rather than a time consuming pre-test. This reduction of overhead provides a force-multiplying effect when combined with validated training systems, as individual lesson plans can be reliably calculated and recalculated in real time, saving time, money, increasing engagement and, presumably, creating better outcomes.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

In this paper we demonstrate the impact of an iterative ADDIE process within GIFT, and furthermore discuss the potential of an expanded ADDIE that allows GIFT to validate pedagogy, tutors, and concept models. Additionally, we explore the role of GIFT as a component in an SOA, and its importance as a source of data for analysis and evaluation. Finally, we make recommendations for potential future development that will improve GIFT's role in an SOA.

GIFT-SEM is the first step to closing the ADDIE loop in GIFT. It provides the ability to evaluate, but is limited in its evaluation to the concept model. In order to fully support the ADDIE process, GIFT needs a more robust set of evaluation modules that are able to evaluate tutor content, pedagogy, and sequencing. To support these modules, we recommend using the GIFT extraction/interpretation layer that has been built as part of GIFT-SEM. This layer currently ingests GIFT logs and extracts data from them in a machine and algorithm actionable format. In the future, this layer could be expanded to also ingest xAPI or other data coming into GIFT from other components of a SOA.

Building on the foundation of the ADDIE loop in GIFT, we foresee expanding ADDIE to include data interoperability and standards alignment that will allow GIFT to evaluate tutors in the context of other systems based on both GIFT’s learner outcome data and learner information gathered from the SOA’s ecosystem. Validated tutors, pedagogy, and other elements of GIFT tutors are the cornerstone of GIFT’s role in a SOA ecosystem. By providing tutors that have gone through design iterations and have been validated against student outcomes, the GIFT output serves as a reusable repository of best practices and existing tutors for the entire SOA. To seamlessly integrate GIFT into large SOA ecosystems, the many powerful, well researched and validated components must each be accessible by the other parts of the SOA.

Future research will determine how GIFT can best function as the central SOA authority, how access protocols should be technically defined, and how services can be written to provide these capabilities to other systems. Finally, we believe that this future research will uncover new ways that GIFT can utilize and be utilized by components of the SOA to provide better pedagogy that is more relevant to learners and provide the best possible learner outcomes.

REFERENCES

ADL. (2017). Total Learning Architecture (TLA). Retrieved January 20, 2017, from Advanced Distributed Learning initiative: https://www.adlnet.gov/tla/

Branch, R. M. (2009). Instructional Design: The ADDIE Approach. DOI:10.1007/978-0-387-09506-6

Dumanoir, P., Bergenthal, J., & Drake, D. (2015). The Live-Synthetic Training, Test and Evaluation Enterprise Architecture. I/ITSEC. NTSA.

Lavendelis, E. (2012). Distributed intelligent tutoring system architectures (pp. 241-266). INTECH Open Access Publisher.

Molenda, M. (2003, 05). In search of the elusive ADDIE model. Performance Improvement, 42(5), 34-36. doi:10.1002/pfi.4930420508

Papazoglou, M. P. (2003, December). Service-oriented computing: Concepts, characteristics and directions. In Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on (pp. 3-12). IEEE.

Sottilare, R. A., Brawner, K. W., Goldberg, B. S., & Holden, H. K. (2012). The generalized intelligent framework for tutoring (GIFT).

Ray, F., Brawner, K., & Robson, R. (2014, July). Using data mining to automate ADDIE. In Educational Data Mining 2014.

Bauer, L., Bravo-Lillo, C., Fragkaki, E., & Melicher, W. (2013, November). A comparison of users' perceptions of and willingness to use Google, Facebook, and Google+ single-sign-on functionality. In Proceedings of the 2013 ACM workshop on Digital identity management (pp. 25-36). ACM.

USAASC. (2017). Synthetic Training Environment (STE). Retrieved January 14, 2017, from http://asc.army.mil/web/portfolio-item/synthetic-training-environment-ste/#

Wang-Costello, J., Goldberg, B., Tarr, R.W., Cintron, L.M., & Jiang,H (2013). Creating an advanced pedagogical model to improve intelligent tutoring technologies. In The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

ACKNOWLEDGEMENTS

The research described herein has been sponsored by the U.S. Army Research Laboratory. The statements and opinions expressed in this article do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

ABOUT THE AUTHORS

Mr. Elliot Robson *is the VP of Research and Operations at Eduworks Corporation and Principal Investigator on GIFT-SEM. He is an expert in data analytics and education with particular expertise in analysis, extracting meaning, and data structures.*

Mr. Ronald "Fritz" Ray *is the Chief Architect at Eduworks Corporation, an expert in GIFT, and designed GIFT-SEM. His interests are in software and data modeling, interoperability, reuse, and security. In his spare time, he replaces specialized hardware devices in his house with Raspberry Pis.*

Dr. Anne M. Sinatra *is an Adaptive Tutoring Scientist and Research Psychologist at U.S. Army Research Laboratory. She is the team modeling vector lead on the GIFT project's Adaptive Training Team. Her background is in human factors psychology, cognitive psychology, and education.*

Toward Simulated Students for Reinforcement Learning-Driven Tutorial Planning in GIFT

Jonathan Rowe[1], Bob Pokorny[2], Benjamin Goldberg[3], Bradford Mott[2], and James Lester[2]
North Carolina State University[1], Intelligent Automation, Inc.[2], U.S. Army Research Laboratory[3]

INTRODUCTION

A critical feature of intelligent tutoring systems (ITSs) is their capacity to guide and scaffold student learning. Tutorial planners leverage contextual information to determine how pedagogical feedback, hints, and prompts should be tailored to learners at run-time (Woolf, 2008). Recent years have witnessed growing interest in applying reinforcement learning (RL) to devise tutorial planners. RL provides a data-driven framework for creating tutorial planners from observations of student behavior and learning outcomes. RL techniques introduce the potential for ITSs that can automatically refine and improve their pedagogical methods over time. RL methods account for the inherent uncertainty in how learners respond to different types of tutorial strategies and tactics, and they produce models that can be automatically induced to optimize measures of student learning. Recent work on reinforcement learning-based tutorial planning has outlined a path for devising pedagogical models across a broad range of learning environments and educational domains (Chi, VanLehn, Litman, & Jordan, 2011; Rowe & Lester, 2015; Williams et al., 2016).

An important challenge for RL-based intelligent tutoring systems is the availability of sufficient data to train pedagogical decision-making models. Advanced learning technologies, such as simulations (Mislevy et al., 2014) and digital games (Rowe & Lester, 2015; Shute, Ventura, & Kim, 2013) often present vast action and state spaces, which raise tractability issues for computational models of tutorial planning. Several projects have trained RL-based intelligent tutors using data from human students, but these systems typically rely upon highly constrained state representations and action sets. An alternate approach is leveraging synthetic data generated by simulated students, which can provide large volumes of training data for RL systems. By simulating students' learning and behavior processes, it is possible to generate effectively unlimited synthetic data for training RL-based tutorial planners. However, this approach presents its own set of challenges, including questions about model granularity, validity, complexity, and efficiency.

In this paper, we survey different approaches for creating simulated students and examine their potential for training RL-based intelligent tutors. We describe steps we are taking to leverage simulated students to induce a modular RL-based tutorial planner for counterinsurgency (COIN) training in GIFT. Specifically, we are investigating the design of simulated students for two complementary COIN training environments: UrbanSim and Urbansim Primer. We discuss major considerations in the design of simulated students for our domain, and we conclude with a discussion of potential enhancements to GIFT that would support the creation of intelligent tutoring systems from simulated student data.

REINFORCEMENT LEARNING-DRIVEN TUTORIAL PLANNING

Reinforcement learning refers to a family of machine learning tasks that induce software control policies for sequential decision-making under uncertainty with delayed rewards (Sutton & Barto, 1998). In classical reinforcement learning, an agent seeks to learn a policy for selecting actions in an uncertain environment in order to accomplish a goal. The environment is characterized by a set of states and a probabilistic model describing transitions between those states. The agent is capable of observing the environment's state and using its observations to guide decisions about which actions to perform. The agent's task is to utilize the reward signal in order to learn a policy that maps observed states to actions and maximizes its total accumulated reward.

Reinforcement learning problems are typically formalized using Markov decision processes (MDPs). An MDP is formally defined by a tuple $M = (S, A, P, R)$, where S is the set of environment *states*; A represents the set of *actions* that the agent can perform; P is the *state transition model*, P: $\{S \times A \times S\} \rightarrow [0, 1]$, which specifies the probability of transitioning to state $s_{t+1} \in S$ after performing action $a_t \in A(s_t)$ in state $s_t \in S$ at time step t; and R is the *reward model*, R: $\{S \times A \times S\} \rightarrow \Re$, which specifies the expected scalar reward $r_t \in \Re$ associated with performing action a_t in state s_t and transitioning to state s_{t+1}. The solution to an MDP is an optimal policy, $\pi^*(s_t) \rightarrow A$, that maps states to actions and yields maximum expected reward for the agent. There are a number of algorithms for solving MDPs under different conditions, including both on-line and off-line contexts.

Reinforcement Learning in Intelligent Tutoring Systems

In our work, we formalize data-driven tutorial planning as a modular reinforcement learning task (Fig. 1). Modular reinforcement learning is a multi-goal extension of classical reinforcement learning that divides a decision-making problem into multiple concurrent sub-problems, each modeled as its own MDP, and machine learning individual policies to solve each sub-problem. Formally, modular reinforcement learning tasks are defined in terms of N concurrent MDPs, $M = \{M_i\}_1^N$, where each $M_i = (S_i, A_i, P_i, R_i)$, corresponding to a sub-problem in the composite reinforcement learning task. Each agent M_i has its own state sub-space S_i, action set A_i, probabilistic state transition model P_i, and reward model R_i. The solution to a modular reinforcement learning problem is a set of N policies, $\pi^* = \{\pi_i^*\}_1^N$, where π_i^* is the optimal policy for the constituent MDP M_i. Whenever two policies π_i and π_j with $i \neq j$ recommend different actions in the same state, an arbitration procedure must be applied. Standard reinforcement learning algorithms can be used to compute solutions for the constituent MDPs. In cases where conflicts occur between concurrent policies for multiple sub-problems, arbitration procedures are employed. Because models for each sub-problem are individually learned, they need only consider those state features, actions and goals that are relevant to the sub-problem.

By decomposing tutorial planning into multiple sub-problems, we can reduce the complexity of reinforcement learning by reframing the task in terms of several smaller, concurrent Markov decision processes, which are solved using modular reinforcement learning methods (Figure 1). To perform this decomposition, we employ the concept of an adaptable event sequence (AES), an abstraction for a recurring series of one or more instructionally related events that, once triggered, can unfold in several different ways within a learning environment. To illustrate the concept of an AES, consider the task of selecting an instructional strategy to deploy after a student has made an

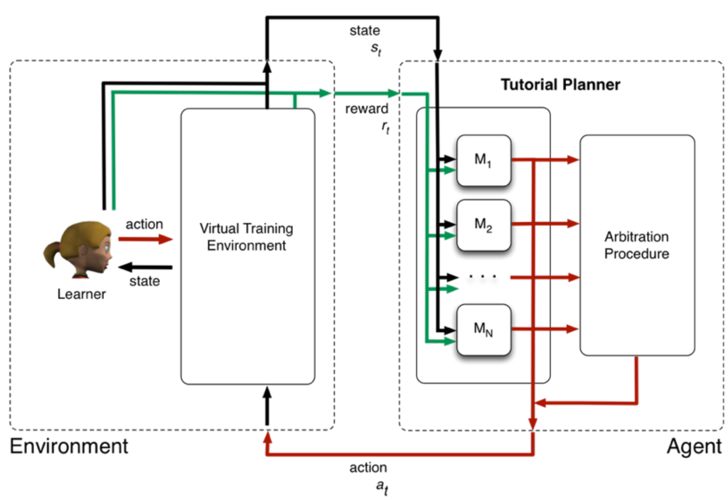


Figure 1. Modular reinforcement learning framework for tutorial planning.

influential decision in a simulation-based learning environment. In our project, the tutorial planner selects among four possible instructional strategies: 1) providing single-topic coaching on a concept that the learner has not yet mastered; 2) providing a multi-topic review on several concepts that the learner has had opportunity to practice; 3) providing feedback on unproductive learning behaviors by the student; or 4) not intervening at all. Each of these four responses is an alternate instructional strategy, and they should be varyingly deployed based on the learner's performance and the state of the training environment. Decisions about what type of instructional strategy to utilize are likely to occur multiple times over the course of a student's learning interaction. Because this tutorial sequence can unfold in one of several valid ways, we refer to it as adaptable, or in other words, it is an adaptable event sequence (AES).

Leveraging the concept of an AES, tutorial planning can be cast as a collection of sequential decision-making problems about scaffolding learning within a virtual training environment. Each AES is modeled as a distinct Markov decision process, M_i. For each AES, every occurrence of the event sequence corresponds to a decision point for M_i. The set of possible scaffolding options for the AES is modeled by an action set, A_i. A particular state representation, S_i, is selected for each AES. State encodes the learner's state and history, as well as the learning environment's state. A state transition model P_i encodes the probability of transitioning between two specific states during successive decision points for the AES. Rewards, R_i, can be calculated from formative or summative assessments of student learning, such as a post-test. Reward is the metric that reinforcement learning is designed to optimize.

To induce optimal policies that solve the MDPs, we can utilize two possible approaches. The first approach is to collect training data from human learners by deploying a tutorial planner that selects actions randomly, in effect sampling the space of tutorial policies and rewards (Chi, VanLehn, Litman, & Jordan, 2011; Rowe & Lester, 2015). Leveraging the mapping between AESs and MDPs, and a training corpus of random tutorial decision data, we can employ model-based reinforcement learning techniques to induce policies for tutorial planning. Typically, dynamic programming methods (e.g., value iteration) are utilized to compute solution policies off-line for each MDP using maximum likelihood estimates of the state transition model and reward model computed from the training corpus (Chi, VanLehn, Litman, & Jordan, 2011; Sutton & Barto, 1998).

An alternate approach is to devise a simulated student that generates synthetic training episodes for on-line reinforcement learning. This involves configuring the RL agent to interface directly with the simulated student; the RL agent provides formal state descriptions and tutorial strategy decisions to the simulated student, and the simulated student returns successor states that emulate state transitions one would expect to observe from actual human students. In addition, the simulated student generates estimates of student learning outcomes, which serve as reward values to drive reinforcement learning. In this manner, an RL-driven tutorial planner can sample different tutorial policies by repeatedly observing the effects of its pedagogical decisions on the simulated student's behavior and learning. As the RL agent samples different policies over many training episodes, it can prioritize more promising areas of the tutorial policy space and ignore areas that are unlikely to yield positive student learning outcomes. In contrast to training a tutorial planner with human student data, the RL agent has access to virtually unlimited training data. It is constrained only by the validity of the simulated student model, as well as the computational resources available for reinforcement learning (e.g., compute cycles, memory).

While training RL-driven tutorial planners with simulated students has several attractive characteristics, creating high-quality simulated students raises a broad range of issues inherent to user modeling. These include selecting the appropriate grain-size for the student model, leveraging an appropriate computational framework, managing the simulation model's complexity, modeling the desired facets of learner behavior, and maintaining the efficiency of the model so that RL can be applied on available computing hardware. In the next section, we survey related work on simulated students, including research on using simulated users for training RL-driven adaptive software systems.

GENERATING SYNTHETIC TRAINING DATA WITH SIMULATED STUDENTS

For several decades, the intelligent tutoring systems community has explored applications of simulated students for adaptive learning environments (VanLehn, Ohlsonn, & Nason, 1993; Beck, Woolf, & Beal, 2000). Recent years have seen growing interest in simulated students, as exemplified by a recurring series of workshops co-located with the International Conference on Artificial Intelligence in Education (AIED-13, AIED-15). An early paper by VanLehn, Ohlsonn, and Nason (1993) outlined three practical applications of simulated students. First, simulated students can be utilized to provide teachers with practice opportunities for refining their instruction. Second, simulated students can serve as co-learners for human students, providing opportunities for collaborative learning. Third, instructional designers can utilize simulated students to conduct formative evaluations of learning materials. In the intervening years, several additional applications have emerged. For example, the SimStudent project has demonstrated that simulated students can be used to author intelligent tutoring systems, using both authoring-by-tutoring and authoring-by-demonstration paradigms (Matsuda, Cohen, & Koedinger, 2014). Another application of simulated students is to generate synthetic data for inducing computational models of intelligent tutoring, which is the subject of our current work (Beck, Woolf, & Beal, 2000; Folsom-Kovarik, Sukthankar, & Schatz, 2013).

Approaches to creating simulated students vary along several dimensions. An especially important dimension is representational grain size. In his seminal work on the Soar cognitive architecture, Allen Newell (1990) described four bands of cognition—biological, cognitive, rational, and social—which organize different time scales for analyzing human action within a unified theory of cognition. Years later, Anderson (2002) revisited these bands, drawing connections between the fine-grained time scales of biology (e.g., milliseconds) and cognitive psychology (e.g., seconds) to coarser grained representations of time scale relevant to education (e.g., 100 hours). Similarly, simulated students operate at varying levels of temporal granularity. Fine-grained simulations have been devised to model human learning at the level of individual knowledge components (Matsuda, Cohen, & Koedinger, 2014). The SimStudent project leverages a production rule representation of procedural knowledge for problem solving in cognitive tutors, which is theoretically based on the ACT-R model of human cognition (Anderson et al., 2004). In contrast, simulated students have been created at the grain size of entire academic programs. The SimGrad project saw the creation of a prototype simulated doctoral program that accounted for courses taken, grades received, enrollment and graduation dates, and other high-level facets of academic performance (Lelei & McCalla, 2015). Most research on simulated students is situated in between these two extremes, focusing on generation of student responses to pedagogical actions at the level of coarse-grained knowledge concepts.

There are also a broad range of computational frameworks for encoding simulated students. Some simulated students are authored as expert systems with hand-crafted models of knowledge and problem solving (VanLehn, Jones, & Chi, 1992). Other simulated students are represented as simple mathematical functions, such as weighted sums (Frost & McCalla, 2015) that have a small number of hand-selected parameters. These models vary in the extent to which they are theoretically grounded, or in same cases, they are based upon the intuitions of the system's designers. Another family of simulation approaches utilize machine learning to induce models of student behavior from corpora of human student data (Beck, Woolf, & Beal, 2000). These models are often effective at capturing the overall statistical distribution of a population of student behaviors, but they risk producing individual episodes that are inconsistent or non-sensible (MacLellan, Matsuda, & Koedinger, 2013).

An important characteristic of student simulations is their degree of model complexity. Model complexity impacts a simulation's capacity to account for individual differences in learning rates, problem-solving strategies, and student traits. VanLehn, Ohlsonn, and Nason (1993) distinguish between (1) tabular simulations and (2) algorithmic simulations. Tabular simulations can be implemented with look-up tables that specify the full range of possible behaviors of the simulated student. In contrast, algorithmic simulations are defined by comparatively complex procedures for generating synthetic student behavior, including behaviors that have not been explicitly encoded by the system's designer. Tabular simulations are efficient to run, straight forward to inspect, and intuitive to author, but they have limited capacity to generalize to unseen situations. Algorithmic simulations vary widely in their efficiency, and examining their behavior requires additional effort, but they are well suited for generalizing to novel situations.

To date, a majority of research on simulated students has focused on cognitive aspects of student learning. This includes simulations that predict students' problem-solving behaviors, learning outcomes, and academic performance (Beck, Woolf, & Beal, 2000; Matsuda, Cohen, & Koedinger, 2014; Rosenberg-Kima & Pardos, 2015). There is comparatively little work investigating simulations of students' emotional, motivational, and metacognitive processes, despite their strong representation in research on user modeling. Sabourin et al. (2013) investigated simulations of students' affective dynamics to investigate whether off-task behavior is an effective emotion regulation strategy in a game-based learning environment for middle school science education. Frost and McCalla (2013) investigated the social effects of peer learners in a simulated environment that recommended personalized sequences of learning objects.

Despite recent interest in applications of RL to education, there are only a small number of examples of simulated students utilized to generate synthetic data for training RL-driven intelligent tutoring systems. The ADVISOR intelligent tutoring system used temporal-difference learning to induce teaching policies from corpora of synthetic student data (Beck, Woolf, & Beal, 2000). The synthetic data were generated by a Population Student Model (PSM), which was comprised of a logistic regression model induced with student data from a series of earlier classroom studies. The PSM generated predictions about how likely a student was to answer a problem correctly and how long the student would take to provide a response. ADVISOR was able to devise teaching policies that reduced human students' problem-solving times and generalized across distinct student populations for a grade school mathematics tutor. More recently, work by Folsom-Kovarik, Sukthankar, and Schatz (2013) devised a partially observable Markov decision process (POMDP) framework for intelligent tutoring, which was evaluated in a Call For Fire task for the U.S. Marine Corps. The framework introduced several techniques for improving the tractability of POMDP-based intelligent tutoring, including *state queues* and *observation chains.* Folsom-Kovarik et al. (2013) utilized simulated students to evaluate different POMDP representations for the underlying learner model.

Since there is a dearth of research on simulated students in RL-driven intelligent tutoring systems, related work on simulated users for intelligent interfaces is of considerable interest. Notably, simulated users serve a key role in RL-driven spoken dialogue systems, and they are particularly important in dialogue management (Schatzmann, Weilhammer, & Young, 2006). Early work on simulated users for spoken dialogue systems was typically linguistically motivated and relied heavily on knowledge-based formalisms (Schatzmann et al., 2006). Similar to educational applications, hand-authoring parameter values in dialogue management is challenging, because human dialogue behavior is often ill understood, population-dependent aspects of dialogue are difficult to estimate, and human authoring is rife with bias. Contemporary research on simulated users for dialogue systems is largely statistical, and it focuses on modeling users with machine learning techniques. Statistical models such as n-grams, Bayesian networks, and hidden Markov models have shown considerable promise for devising effective user simulations for training RL-driven dialogue managers (Schatzmann et al., 2006).

TRAINING TUTORIAL PLANNERS WITH SIMULATED STUDENTS IN GIFT

We are investigating data-driven tutorial planning in GIFT for two COIN training environments: (1) the UrbanSim Primer hypermedia-based learning environment, and (2) the UrbanSim simulation-based learning environment. These two learning environments are complementary. Typically, learners will complete several units of the UrbanSim Primer to familiarize themselves with foundational COIN concepts, and afterward they will complete a series of training scenarios in the UrbanSim simulation. Both of these learning environments have been integrated with GIFT in order to serve as a general testbed for creating and evaluating RL-driven tutorial planners for COIN training.

The UrbanSim Primer is a hypermedia-based learning environment that provides direct instruction on complex counterinsurgency and stability operations. Developed by the USC Institute for Creative Technologies, the UrbanSim Primer presents hyperlinked video, audio, text, and diagrams on a range of doctrinal concepts of counterinsurgency, including the importance of population support, strategies such as Clear-Hold-Build, resources and processes for intelligence gathering, and issues in successful execution of COIN operations. The Primer also provides preliminary instruction on the usage of the UrbanSim simulation.

UrbanSim is an open-ended simulation-based virtual training environment for counterinsurgency and stability operations (Fig. 2). In UrbanSim, learners act as a battalion commander whose mission is to maximize civilian support for the host nation government. Training experiences using UrbanSim resemble computer gameplay interactions with turn-based strategy games. On each turn, the learner assigns actions for 11 Battalion resources, such as “E Company, A platoon patrols the Malmoud Quarter” or “G Company, B platoon recruits policemen in the Northern Area.” Trainees’ actions, and consequences to their actions, are simulated using an underlying social-cultural behavior engine that determines how the host city’s inhabitants respond to different situations.

Generalized Instructional Strategies for COIN Training

We selected four types of instructional strategies for modeling and delivery by RL-driven tutorial planners for COIN training. The instructional strategies enable a common encoding of pedagogical actions across both the UrbanSim and UrbanSim Primer learning environments (Figure 2). The instructional strategies include (1) single-topic coaching, (2) multi-concept review, (3) feedback on unproductive learning behaviors, and (4) no feedback. These instructional strategies are delivered to learners using GIFT’s Tutor User Interface (TUI).

Single-topic coaching consists of a text-based feedback message about a specific dimension of learner performance in either the *security* or *meetings with host-nation leaders* performance areas of COIN training. These feedback messages can be delivered at the end of any turn in UrbanSim, or alternatively, at the end of an UrbanSim Primer unit, which is presented in the form of a PowerPoint show. Multi-concept reviews are similar to single-topic coaching, except that they address multiple dimensions of

Figure 2. UrbanSim simulation-based training

COIN performance simultaneously. Multi-concept reviews interleave summaries of effective COIN operational practice and excerpts from relevant U.S. Army field manuals, such as the *Commander's Handbook for Strategic Communication and Communication Strategy*. Feedback on unproductive learning behaviors focuses on addressing egregious or inefficient actions performed by learners that have little or no relevance to the learning task within UrbanSim or the UrbanSim Primer.

For each of these instructional strategies, we select among three possible variants for implementing the strategy. The variants are based upon the ICAP framework (Chi, 2009), which distinguishes between (1) interactive, (2) constructive, (3) active, and (4) passive forms of instructional activities. Our project does not focus on "interactive" forms of instructional methods, which typically refer to tutorial dialogues, so we have devised instructional strategies consistent with constructive, active, and passive forms of each technique. In other words, each of the three instructional techniques in this project—single-topic coaching, multi-concept reviews, and unproductive learning behaviors—has passive, active, or constructive variants.

The passive form of an instructional technique consists solely of a text-based message that participants read prior to continuing with their training. Passive instructional strategies do not require a particular response from the learner beyond clicking a button at the conclusion of the feedback message, but they are efficient and enable learners to promptly return to hypermedia or simulation-based training. The active form of an instructional technique expands upon the passive strategy by prompting learners to highlight key parts of feedback, or review the message, to identify its most important elements. After the learner completes her highlight, she is presented with an expert highlight of the same instructional message in order to facilitate critical evaluation of her own active learning performance. The constructive form of an instructional technique expands further by prompting learners to briefly summarize, in their own words, the most important parts of the feedback or review message. After the learner finishes writing her summary, she is presented with an expert summary in order to facilitate her own evaluation of her learning performance.

Designing Generalized Simulated Students for COIN Training Environments

In order to create data-driven tutorial planners for COIN training, we are currently devising simulated students to emulate behavior patterns and learning outcomes of human students interacting with the training system. We have a small dataset from an initial pilot test (N=23) conducted with ROTC cadets using UrbanSim and the UrbanSim Primer, which is informing efforts to manually author coarse-grained student simulations for each of the two learning environments. Both sets of student simulations are encoded in tabular format as probability mass functions; their format is closely related to the state-transition and reward models specified in MDP models of tutorial planning. This format was chosen because it is sufficiently granular to provide synthetic data for reinforcement learning, and it is highly efficient for generating large volumes of synthetic data. Devising a more fine-grained simulation is beyond the scope of the project, because granular cognitive-task analyses have not been conducted for UrbanSim and UrbanSim Primer.

Each simulated student is designed as a *bipartite model*. First, it consists of a joint probability distribution characterizing stochastic transitions between tutorial planner states. Second, it includes a joint probability distribution characterizing student learning outcomes from terminal states. The probability values in these models are informed by aggregated observations of state transitions and learning outcomes from the pilot test data. However, data sparsity issues require manual estimation of missing probability values. These model parameters will be validated and refined as additional data is collected from human students during the project. For UrbanSim, the temporal grain size for a simulated student corresponds to a single turn of the training simulation. For UrbanSim Primer, the grain size corresponds to a single lesson, which is typically a few minutes in duration. The student simulations focus on modeling cognitive and behavioral facets of student learning.

We model high-level decisions about pedagogical strategies in terms of three binary AESs: a single-topic coaching AES, a multi-concept review AES, and an unproductive behavior feedback AES. Each of these AESs is modeled by a distinct MDP, and similarly, it is associated with its own bipartite simulated student model. In addition, each of the aforementioned AESs is associated with a lower-level AES that encodes decisions about ICAP-inspired implementation strategies. In other words, if the tutorial planner chooses to deliver single-topic coaching, the planner's control flow transitions to a follow up MDP that selects among passive, active, and constructive variants of the coaching intervention. If the tutorial planner chooses to deliver a multi-concept review, the planner's control flow moves to a different MDP for passive, active, constructive decisions.

The state features for simulated students draw upon several sources: (1) student mastery of relevant knowledge concepts, (2) relevant task states, (3) learner attributes, and (4) pedagogical history. Because the MDP models interface primarily with GIFT's Pedagogical Module, their state representations are restricted to domain-independent features. The same is true of state representations for the simulated students. The selection of specific state features for the simulated students is ongoing, but the choice of features will seek to balance between model complexity and expressiveness. The actions that each simulated student responds to are the pedagogical actions associated with each AES. Rewards will correspond to discretized COIN content learning gains for UrbanSim Primer and simulation training performance for UrbanSim.

In practice, each simulated student will be instantiated with several different configurations of parameters, allowing the simulated student model to reflect a population of student learners, rather than behaviors of a single student. We intend to investigate the effects of alternate parameterizations of simulated students on the learning rates and policies yielded for the RL-driven tutorial planner. In addition, we intend to qualitatively analyze the resulting tutorial policies in light of current theory on instructional design and learning science.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

Data-driven approaches to tutorial planning, such as reinforcement learning, show significant promise for devising effective models of instructional strategies for complex domains and learning environments. We are investigating RL-driven tutorial planning in the domain of COIN training, with a focus on the UrbanSim Primer and UrbanSim learning environments. We have presented a brief survey of the research literature on simulated students, which suggests that model granularity, choice of computational framework, model complexity, modeled learning behavior, and efficiency are key factors that distinguish different types of simulated students. Leveraging findings from this review, we are currently devising generalized simulated students for COIN training, which will be used to generate synthetic data for training RL-driven tutorial planners in GIFT.

There are several promising directions for future work. Conducting studies to validate simulated students by comparing synthetic data with actual human behaviors will be an important step. Devising tools, workflows, and examples for incorporating tutorial planning policies induced from simulated students in GIFT is planned. In addition, providing tools for non-experts to work with simulated students, including creating, configuring, sharing, and refining simulated student models, holds potential to significantly expand research on simulated students and advance GIFT's objectives of realizing low cost, automatically generated instruction across a broad range of training domains.

ACKNOWLEDGMENTS

The research described herein has been sponsored by the U.S. Army Research Laboratory under cooperative agreement W911NF-15-2-0030. The authors would like to thank the faculty and students of the NC State University Army ROTC program for their assistance this research. The statements and opinions expressed in this article do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

REFERENCES

Anderson, J. R. (2002). Spanning seven orders of magnitude: A challenge for cognitive modeling. *Cognitive Science*, 26, 85-112.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of mind. *Psychological Review*, *111*(4), 1036-1060.

Beck, J., Woolf, B. & Beal, C. (2000). ADVISOR : A machine learning architecture for intelligent tutor construction. *Proceedings of the 17th National Conference on Artificial Intelligence*, Austin, TX, pp. 552–557.

Chi, M.T.H. (2009). Active-Constructive-Interactive: A conceptual framework of differentiating learning activities. *Topics in Cognitive Science, 1*(1), 73-105.

Chi, M., VanLehn, K, Litman, D. & Jordan, P. (2011). An evaluation of pedagogical tutorial tactics for a natural language tutoring system: A reinforcement learning approach. *International Journal of Artificial Intelligence in Education*, 21(1-2), 83–113.

Folsom-Kovarik, J.T., Sukthankar, G. & Schatz, S. (2013). Tractable POMDP representations for intelligent tutoring systems. *ACM Transactions on Intelligent Systems and Technology*, 4(2), 1–22.

Frost, S. & McCalla, G. (2015). An Approach to Developing Instructional Planners for Dynamic Open-Ended Learning Environments. *Proceedings of the Workshops at the 17th International Conference on Artificial Intelligence in Education (AIED-15)*, Madrid, Spain, pp. 1–10.

Lelei, D. E. & McCalla, G. (2015). Exploring the Issues in Simulating a Semi-Structured Learning Environment: The SimGrad Doctoral Program Design. *Proceedings of the Workshops at the 17th International Conference on Artificial Intelligence in Education (AIED-15)*, Madrid, Spain, pp. 11–20.

MacLellan, C., Matsuda, N., & Koedinger, K. (2013). Toward a Reflective SimStudent: Using Experience to Avoid Generalization Errors. *Proceedings of the AIED 2013 Simulated Learners Workshop*. Memphis, TN, pp. 51–60, Springer.

Matsuda, N., Cohen, W.W., & Koedinger, K.R. (2014). Teaching the Teacher: Tutoring SimStudent Leads to More Effective Cognitive Tutor Authoring. *International Journal of Artificial Intelligence in Education*, 25(1), 1–34.

Mislevy, R. J., Oranje, A., Bauer, M. I., von Davier, A., Hao, J., Corrigan, S., Hoffman, E., DiCerbo, K., & John, M. (2014). *Pyschometric Considerations in Game-Based Assessments*. Retrieved from https://www.ets.org/research/policy_research_reports/publications/white_paper/2014/jrrx

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University.

Rosenberg-Kima, R. & Pardos, Z., 2015. Is this Model for Real? Simulating Data to Reveal the Proximity of a Model to Reality. *Proceedings of the Workshops at the 17th International Conference on Artificial Intelligence in Education (AIED-15)*, Madrid, Spain, pp. 78–87.

Rowe, J. & Lester, J. (2015). Improving Student Problem Solving in Narrative- Centered Learning Environments: A Modular Reinforcement Learning Framework. *Proceedings of the 17th International Conference on Artificial Intelligence in Education*, Madrid, Spain, pp. 419-428.

Sabourin, J., Rowe, J., Mott, B., & Lester, J. (2013). Considering Alternate Futures to Classify Off-Task Behavior as Emotion Self-Regulation: A Supervised Learning Approach. *Journal of Educational Data Mining*, 5(1), 9–38.

Schatzmann, J., Weilhammer, K., Stuttle, M., & Young, S. (2006). A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The Knowledge Engineering Review*, *21*(2), 97–126.

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.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.

VanLehn, K., Jones, R. M., & Chi, M. T. (1992). A model of the self-explanation effect. *The Journal of the Learning Sciences*, 2(1), 1–59.

VanLehn, K., Ohlsson, S. & Nason, R. (1993). Applications of Simulated Students: An Exploration. *Journal of Artificial Intelligence in Education*, 5(2), 1–42.

Williams, J. J., Kim, J., Rafferty, A., Maldonado, S., Gajos, K., Lasecki, W., & Heffernan, N. (2016). AXIS: Generating Explanations at Scale with Learnersourcing and Machine Learning. *Proceedings of the 3rd Annual ACM Conference on Learning at Scale*, Edinburgh, SCT, pp. 379-388.

Woolf, B. P. (2008). *Building Intelligent Interactive Tutors: Student-Centered Strategies for Revolutionizing e-Learning*. Morgan Kaufmann.

ABOUT THE AUTHORS

__Dr. Jonathan Rowe__ is a Research Scientist in the Center for Educational Informatics at North Carolina State University. He received the Ph.D. and M.S. degrees in Computer Science from North Carolina State University, and the B.S. degree in Computer Science from Lafayette College. His research is in the areas of artificial intelligence and human-computer interaction for advanced learning technologies, with an emphasis on game-based learning environments, intelligent tutoring systems, user modeling, educational data mining, and computational models of interactive narrative.

__Dr. Bob Pokorny__ is Director of the Education and Training Technology Division at Intelligent Automation, Inc. He earned his Ph.D. in Experimental Psychology at University of Oregon in 1985, and completed a postdoctoral appointment at University of Texas at Austin in Artificial Intelligence. Bob's first position after completing graduate school was at the Air Force Research Laboratory, where he developed methodologies to efficiently create intelligent tutoring systems for a wide variety of Air Force jobs. At Intelligent Automation, Bob has led many cognitive science projects, including adaptive visualization training for equipment maintainers, and an expert system approach for scoring trainee performance in complex simulations.

__Dr. Benjamin Goldberg__ is a member of the Learning in Intelligent Tutoring Environments (LITE) Lab at the U.S. Army Research Laboratory's (ARL) Human Research and Engineering Directorate (HRED), Simulation and Training Technology Center (STTC) in Orlando, FL. He has been conducting research in modeling and simulation for the past five years with a focus on adaptive learning and how to leverage artificial intelligence tools and methods for adaptive computer-based instruction. Currently, he is the LITE Lab's lead scientist on instructional strategy research within adaptive training environments. Dr. Goldberg is a Ph.D. graduate from the University of Central Florida in the program of Modeling & Simulation.

__Dr. Bradford Mott__ is a Senior Research Scientist in the Center for Educational Informatics at North Carolina State University. Prior to joining North Carolina State University, he served as Technical Director at Emergent Game Technologies where he created cross-platform middleware solutions for video game development, including solutions for the PlayStation 3, Wii, and Xbox 360. Dr. Mott received his Ph.D. in Computer Science from North Carolina State University in 2006, where his research focused on intelligent game-based learning environments. His current research interests include computer games, computational models of interactive narrative, and intelligent game-based learning environments.

Dr. James Lester *is Distinguished Professor of Computer Science at North Carolina State University, where he is Director of the Center for Educational Informatics. His research centers on transforming education with technology-rich learning environments. With a focus on adaptive learning technologies, his research spans intelligent tutoring systems, game-based learning environments, affective computing, and tutorial dialogue. The adaptive learning environments he and his colleagues develop have been used by thousands of students in K-12 classrooms. He received his Ph.D. in Computer Science from the University of Texas at Austin in 1994. He is a Fellow of the Association for the Advancement of Artificial Intelligence (AAAI).*

THEME VI: DOMAIN MODELING

Expanding Domain Modeling in GIFT

Robert A. Sottilare, Ph.D.
US Army Research Laboratory – Human Research & Engineering Directorate (ARL-HRED)
Learning in Intelligent Tutoring Environments (LITE) Lab
Center for Adaptive Instructional Sciences (CAIS)

INTRODUCTION

The purpose of this paper is to update users of the Generalized Intelligent Framework for Tutoring (GIFT; Sottilare, Brawner, Goldberg, & Holden, 2012; Sottilare, Brawner, Sinatra, & Johnston, 2017) on new and emerging capabilities to represent a broader variety of task domains in Intelligent Tutoring Systems (ITSs) in support of adaptive instruction. Adaptive instruction delivers content, offers feedback, and intervenes with learners based on tailored strategies and tactics with the goal of optimizing learning, performance, retention, and transfer of skills for both individual learners and teams. GIFT is a tutoring architecture that has evolved over the last five years with three primary goals: 1) reduce the time and skill required to author ITSs, 2) automate best practices of instruction in the policy, strategies, and tactics of tutoring, and 3) provide a testbed to assess the effectiveness of adaptive instructional tools and methods with respect to learning, performance, retention, and transfer of skills. Another overarching goal for GIFT has been to adapt ITSs to provide instruction in militarily-relevant training and educational domains.

The US Army Learning Model (ALM; U.S. Army Training & Doctrine Command, 2011) notes that training and education tools and methods must be of sufficient intelligence to understand the needs of individual learners and teams, and adapt to mitigate negative learner states, and to guide and tailor instruction in real time to optimize learning, performance, retention, and transfer of skills from instruction to operations. This is the basis of self-regulated learning (SRL) where Soldiers are expected to largely manage their own learning and career development through the growth of metacognitive (e.g., reflection), self-assessment, and motivational skills (Butler and Winne, 1995) with guidance from artificially-intelligent software-based agents. Effective guidance can only come from informed agents who fully understand the states, traits, and limitations of the learner along with subject matter expertise of the domain under training.

Currently, most ITSs are focused on cognitive task domains (e.g., problem solving and decision making) in academic subjects that primarily include software programming, physics, and mathematics. While there are many military task domains that involve cognitive skill development (e.g., military planning processes and assessment of battlespace strategies and tactics), many more involve interdependent team processes (e.g., building clearing) and psychomotor skills (e.g., marksmanship). It is for this reason that we desire to extend current capabilities in GIFT to support content delivery, assessment, and remediation processes for more complex team and psychomotor tasks. The following section describes some of the challenges to expanding domain modeling beyond cognitive tasks and beyond the current model of desktop training.

CHALLENGES IN EXPANDING DOMAIN MODELING

As GIFT has been designed to be largely domain-independent except for the domain model, the concept of domain modeling is vital. Research in domain modeling strives to make GIFT generalizable for multiple types of tasks (cognitive, affective, psychomotor, and social) and provides flexibility to facilitate the reuse of content and structure. In 2015, Sottilare, Sinatra, Boyce, & Graesser documented domain modeling goals/challenges and approaches. Goals/challenges follow:

- Understand and model the characteristics, similarities, and differences of military training domains (cognitive, affective, psychomotor, social, and hybrid) with respect to their associated knowledge representations to support more efficient and effective authoring, instruction, and evaluation of adaptive training tools and methods
- Understand and model the dimensions (definition, complexity, and dynamics) of training domain representations to extend the capabilities of traditional ITSs; thereby, supporting challenging, militarily-relevant training domains

Below are research approaches to modeling domain content and dimensions:

- Examine the efforts required to author domains of varying complexity, definition, and physical dynamics and identify methods
- Define methods to measure task domain complexity to allow comparative evaluation of different authoring systems and capabilities
- Examine domains for ill-defined and well-defined tasks to understand differences and support authoring processes for both
- Examine the composition of militarily-relevant training and education domains across the spectrum of cognitive, affective, psychomotor, and social tasks to understand requirements for authoring
- Discover/examine methods to match the nature of military tasks in training/educational environments and operational environments to optimize transfer of skills, and evaluate methods to determine the return on investment (ROI) for high levels of compatibility
- Discover methods to accurately assess learning and domain task performance in real-time
- Discover methods to promote optimal learning, performance, retention and transfer (on-the-job performance) across domains
- Discover tools and methods to support individual and team training (e.g., small unit and collective training) and education (e.g., collaborative learning and problem-solving) experiences

If we examine the complexity of tasks, we can see tasks that are trained exactly as they are executed in the operational environment. These tasks are the most dynamic and have the greatest chance to transfer skills from training to operations. Tasks where there is less of a match between training actions and operational actions have a lower opportunity for transfer, but are also less complex and therefore less expensive to build. Before we begin examining new and emerging domains, it is useful to the following hierarchy helps define complexity based on task dynamics:

- static training (e.g., desktop training), lower complexity, lower transfer potential; more cognitive
- limited dynamic (e.g., adaptive marksmanship training), limited movement, moderate transfer potential, mix of cognitive and physical
- enhanced dynamic (multi-learner tasks in instrumented spaces), operational movement in a restricted space, moderate to high transfer potential, mix of cognitive and physical
- in-the wild (instrumented learners), operational movement in an unrestricted space, high transfer potential, high degree of physical dynamics

The following sections describe areas of new or emerging capabilities to support the goal of expanding GIFT to a wider variety of task domains.

TUTORING MARKSMANSHIP: A PSYCHOMOTOR TASK DOMAIN

The most mature psychomotor domain in terms of research and development of a working prototype is marksmanship. GIFT now has a coordinated set of sensors that identify behaviors that are critical to successful marksmanship. The prototype has now been integrated with PEO STRI's Engagement Skills Trainer to demonstrate interaction of the learner with stationary targets, assessment of the learner's performance, and remediation of any detected errors by the tutor as shown in Figure 1.

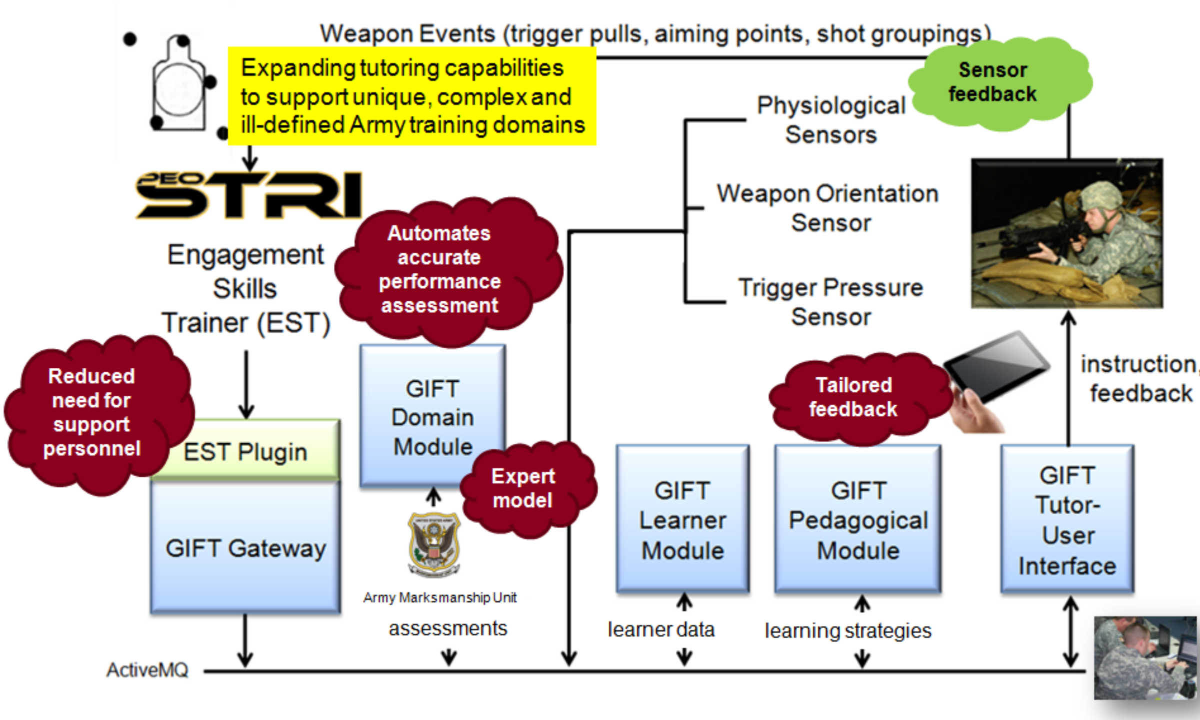


Figure 1. Interaction between learner, marksmanship environment, and the ITS.

TUTORING MEDICAL TRIAGE AND HEMORRHAGE CONTROL

The adaptive instruction provided by Intelligent Tutoring Systems (ITSs) tailors direction, support, and feedback to enhance/maintain the learning needs (e.g., lack of knowledge or skill) of each individual. As noted earlier, ITSs are generally developed to support desktop instructional applications involving cognitive problem solving and decision-making tasks. Recently, GIFT has been used to provide tailored training military tasks using desktop applications (e.g., Virtual Battlespace and Virtual Medic). The degree of transfer of skills from training to operations is limited since training is more focused on the process and much less on the interaction between the learner and the virtual patient. For this reason, the military establishment requires adaptive instruction to extend beyond the desktop to be compatible with the physical nature of many tasks encountered.

In 2016, Sottilare, Hackett, Pike & Laviola examined how commercial sensor technologies might be adapted to work with GIFT and support tailored computer-guided instruction in the psychomotor domain for a military medical training task, specifically hemorrhage control. Toward this goal, they evaluated the usability and system features of commercial smart glasses and pressure-sensing technologies. Smart glasses were selected as the focus of this study over handheld mobile devices in order to promote a hands-free experience during the training of hemorrhage-control tasks on a mannequin. Pressure sensors were selected to provide direct measures of effectiveness during the application of tourniquets and pressure bandages in controlling blood flow. Their findings demonstrated the feasibility of using commercial technology to train hemorrhage control. Smart glasses could provide visual effects (e.g., wounds and bleeding) while pressure sensors could be directly integrated into tourniquets and bandages to relay data about wound pressure. A next step is to build a prototype and begin testing limitations (e.g., distance between pressure sensors and computational platforms (e.g., computers or smartphones)).

TUTORING SPORTS: PSYCHOMOTOR TASKS AND BREATHING

This year (2017) Kim, Dancy, Goldberg, & Sottilare asked the question: does tactical breathing during a psychomotor task influence skill development while under adaptive instruction? Tactical breathing is a specific breath-control technique used by individuals to perform a precision action required psychomotor task in a stressful environment (Neumann & Thomas, 2009; Neumann & Thomas, 2011). The focus of this research is to examine the relationship between cognitive (e.g., attentional resources) and physiological (e.g., breathing) factors during the execution a psychomotor task (i.e., golf putting). It is not well understood that how the corresponding mechanisms of attentional control interact with the physiological factors as the learner progresses to the learning stage. If attentional capacity changes over time during the learning stage, an adaptive instructional system such as a GIFT-based tutor could provide tailored feedback to the learner to refocus their attentional resources. Next steps in this effort are to experimentally examine the relationship between attentional resources and a broader set of physiological factors in a stressful task environment.

TUTORING IN THE WILD: AUGMENTED REALITY ENVIRONMENTS

Intelligent Tutoring Systems (ITSs) have been shown to be effective in training tools for a variety of military tasks. However, these systems are often limited to controlled laboratory settings in which they exercising cognitive skills (e.g., decision-making and problem solving) at on desktop or laptop computers. These tools may potentially limit the learning and retention of military members who are training to master physical tasks or tasks with physical aspects (e.g., psychomotor tasks). Augmented reality, mostly real with virtual effects, presents the possibility of combining intelligent tutoring with hands-on applications in realistic physical environments. Sottilare & LaViola (2015) and LaViola, et al (2015) began to examine the use of an augment-reality based adaptive tutoring system for instruction in the wild, locations where no formal training infrastructure is present. One of their goals was to identify the challenges of transitioning from desktop tutoring to the wild. Another was to examine low cost commercial smartglasses to understand their benefits and limitations. Virtual humans and virtual objects were placed in various locations within the lab. They found it was feasible to employ AR as a tutoring tool in a restricted laboratory environment in order to control lighting/contrast, the persistence of the environment, and power consumption. Vargas (2017, *in press*) began to examine how to author (create and place) virtual humans and objects in AR environments. Next steps are to evaluate what it will take to make the system portable for use in a variety of lighting conditions.

TUTORING TEAM TASKS: TEAMWORK AND TASKWORK

Currently, there are no tools or methods available in the public baseline for modeling or tutoring teams in GIFT. However, there are research initiatives focused on team modeling, and identification of teamwork and taskwork processes. The major goals/challenges for modeling teams of learners are similar to those for individual learners. In 2015, Goodwin et al documented team modeling goals/challenges and approaches. Identified goals/challenges follow:

- Real-time acquisition of team behavioral measures for application in machine learning classifiers
- Real-time classification of collective taskwork and teamwork states to support adaptive instructional decisions in complex environments
- Classification of team competency using long term individual team member data (e.g., achievements, demographics, traits) stored in learning management systems and individual record stores
- Maintaining the accuracy of classification methods in environments with data issues (e.g., small samples, missing or ill-defined data) and within complex systems
- Support of team instruction in militarily-relevant team task domains (e.g., building clearing, collaborative problem solving)
- Lack of capability to handle and process large amounts of structured and unstructured team data (also referred to as big data)
- Lack of an easily accessible, persistent, cost-effective, and low-overhead training environment for teams of learners

Below are research approaches to acquiring team data and accurately classifying team states:

- Evaluate the performance of unobtrusive sensors in dependably acquiring team behavioral data
- Evaluate the performance (accuracy) of machine learning classifiers for various states related to teamwork and collective taskwork performance
- Examine and validate the accuracy of semantic analysis and other classification techniques in classifying/predicting domain competency of teams based on their collective experiences/achievements
- Examine reinforcement machine learning techniques to continuously improve instructional strategy and tactic selection for team training and educational experiences
- Examine machine learning techniques for working with small samples, missing data or inaccurate data for teams of learners
- Examine opportunities to link GIFT Cloud with external individual training simulations and serious games to provide an easily accessible, persistent, cost-effective, low-overhead training environment for adaptive team instruction

One way of extending domain-independence to the modeling of teams is to separate domain-independent teamwork behaviors from task-specific, domain-dependent behaviors. Salas (2015) distinguishes teamwork, interactions between team members, from taskwork, behaviors demonstrated in executing the task. An examination of teamwork activities (e.g., coaching or conflict management) via a meta-analysis of the team training and performance literature led to the identification of several behavior markers for high performing teams (Sottilare, et al, 2017, *in review*). Next steps are to seek methods to unobtrusively acquire these behavioral markers in order to identify team states and subsequently assign the ITS to manage them.

ACKNOWLEDGMENTS

The research described herein has been sponsored by the U.S. Army Research Laboratory. Statements and opinions expressed in this paper do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

REFERENCES

Butler, D. L., & Winne, P. H. (1995). Feedback and self-regulated learning: A theoretical synthesis. *Review of educational research*, *65*(3), 245-281.

Goodwin, G., Johnston, J., Sottilare, R., Brawner, K., Sinatra, A., & Graesser, A. (2015). Individual learner and team modeling for adaptive training and education in support of the US Army Learning Model: Research Outline (No. ARL-SR-0336). ARMY RESEARCH LAB ABERDEEN PROVING GROUND MD HUMAN RESEARCH AND ENGINEERING DIRECTORATE.

Kim, J., Dancy, C., Goldberg, B., & Sottilare, R. (2017; *in press*). A Cognitive Modeling Approach - Does Tactical Breathing in a Psychomotor Task Influence Skill Development during Adaptive Instruction? In *Foundations of Augmented Cognition* (pp. xx-xx). Springer International Publishing.

LaViola, J., Sottilare, R., Garrity, P., Williamson, B., Brooks, C., and Veazanchin, S. (2015). Using Augmented Reality to Tutor Military Tasks in the Wild. In Proceedings of the *Interservice/Industry Training Simulation & Education Conference*, Orlando, Florida, December 2015.

Neumann, D.L. & Thomas, P.R. (2009). The relationship between skill level and patterns in cardiac and respiratory activity during golf putting. Int. J. Psychophysiol. 72(3), 276–282.

Neumann, D.L. & Thomas, P.R. (2011). Cardiac and respiratory activity and golf putting performance under attentional focus instructions. Psychol. Sport Exerc. 12(4), 451–459.

Salas, E. (2015). Team Training Essentials: A research-based guide. Routledge Publishing. New York & London.

Sottilare, R. and LaViola, J. (2015). Extending Intelligent Tutoring Beyond the Desktop to the Psychomotor Domain: A survey of smart glass technologies. In Proceedings of the *Interservice/Industry Training Simulation & Education Conference*, Orlando, Florida, December 2015.

Sottilare, R.A., Brawner, K.W., Goldberg, B.S. & Holden, H.K. (2012). The Generalized Intelligent Framework for Tutoring (GIFT). Concept paper released as part of GIFT software documentation. Orlando, FL: U.S. Army Research Laboratory – Human Research & Engineering Directorate (ARL-HRED). Retrieved from: https://gifttutoring.org/attachments/152/GIFTDescription_0.pdf

Sottilare, R., Hackett, M., Pike, W., and LaViola, J. (2016). Adaptive Instruction for Medical Training in the Psychomotor Domain. In J. Cohn, D. Fitzhugh, and H. Freeman (Eds.) Special Issue: Modeling and Simulation Technologies to Enhance and Optimize the DoD's Medical Readiness and Response Capabilities of the *Journal for Defense Modeling & Simulation (JDMS).*

Sottilare, R., Brawner, K., Sinatra, A., & Johnston, J. (2017). An Updated Concept for a Generalized Intelligent Framework for Tutoring (GIFT). Orlando, FL: U.S. Army Research Laboratory – Human Research & Engineering Directorate (ARL-HRED).

Sottilare, R., Sinatra, A., Boyce, M., & Graesser, A. (2015). Domain Modeling for Adaptive Training and Education in Support of the US Army Learning Model-Research Outline (No. ARL-SR-0325). ARMY RESEARCH LAB ABERDEEN PROVING GROUND MD HUMAN RESEARCH AND ENGINEERING DIRECTORATE.

Sottilare, R.A., Burke, C.S., Salas, E., Sinatra, A.M., Johnston, J.H. & Gilbert, S.B. (2017). Towards a Design Process for Adaptive Instruction of Teams: A Meta-Analysis. *International Journal of Artificial Intelligence in Education.* DOI: 10.1007/s40593-017-0146-z.

U.S. Army Training & Doctrine Command (TRADOC). (2011). 525-8-2 The US Army Learning Concept for 2015. Washington, DC: Department of the Army.

ABOUT THE AUTHOR

Dr. Robert Sottilare *leads adaptive training research within ARL's Learning in Intelligent Tutoring Environments (LITE) Lab and is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT). He is ARL's technical lead for the Center for Adaptive Instructional Sciences (CAIS).*

Authoring Augmented Reality Scenarios for Intelligent Tutoring Systems with GIFT

Andrés Vargas González[1], Brian Williamson[1], Dr. Joseph J LaViola Jr[1], Dr. Robert Sottilare[2]
[1]University of Central Florida, [2]U.S. Army Research Laboratory

INTRODUCTION

Developments in see-through displays combined with multi-modal interaction techniques (Renambot, et al., 2014) have revealed rich possibilities in the realms of tutoring and training. The use of depth cameras, precise tracking devices and vision algorithms allow for a combination of real world environment training with virtual elements. In this paper we explore this domain by examining the interface needed to couple augmented reality (AR) scenarios with an intelligent tutoring system.

We propose a user interface for authoring an augmented reality tutoring course built upon the Generalized Intelligent Framework for Tutoring, known as GIFT (Sottilare, Brawner, Goldberg, & Holden, 2012). This intelligent tutoring framework contains a broad set of tools and algorithms to author tutoring systems that utilize concepts such as Learning Effect Model (Sottilare R. A., 2015) and can be applied to individuals or in teams. GIFT also provides efficient development of intelligent tutoring systems (ITSs) and is backed by a broad set of components in the learner, domain and instructional modeling (Goodwin, et al., 2015; Sottilare, Ososky, & Boyce, 2015; Goldberg, Sottilare, Moss, & Sinatra, 2016). Combined with the tool we developed, the GIFT Augmented Reality Authoring Tool (GARAT), we show a starting point in analyzing and understanding the problems that arise when integrating an intelligent tutoring system with an augmented reality scene.

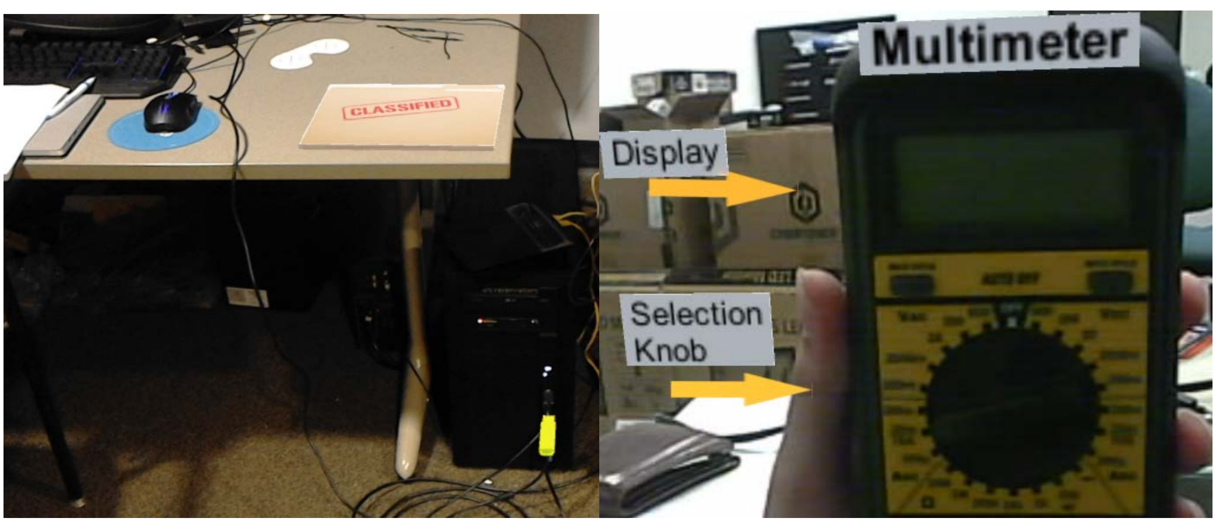


Figure 1. Two example AR courses developed in this paper. The left image shows a training scenario with sensitive information represented by virtual items left unattended. The right image shows training of a multimeter with indicators presented to the user.

We developed the environment that allows to author augmented reality scenarios via GARAT, connect to GIFT via interops and a testbed for the development of tutoring external applications via HoloLens (Statt, 2015). GARAT does not support any other course information authoring but allows to be integrated in the GIFT Authoring Tool pipeline. The GARAT interface works with an external training application running on the latest commercial off the shelf see-through display, the HoloLens, which is used for both the authoring tool and the display of the augmented reality course. While this device is used as the primary interaction medium, our proposed interface can be extended to any platform. The HoloLens creates a 3D mapping of the real world environment which is represented as a mesh back into the authoring tools. While not highly detailed we demonstrate its ability to provide context in the tutoring course design. When the course is executed, the HoloLens's accurate head tracking is used to anchor virtual objects to the real world.

Two example training applications were developed for this paper, both shown in Figure 1. First, we have the user identify possible information risks within a sensitive compartmented information facility (SCIF). The SCIF itself is simulated with virtual items used to represent the identifiable risks. Thanks to the use of augmented reality the training may take place in any environment without creating a real risk of an information leak. The second application involves requesting information regarding a specific device, a multimeter, back to the tutor.

The integration of augmented reality and intelligent tutoring is a new domain which we demonstrate a starting point in the authoring of an AR course and the problems that must be overcome in doing so. We also show a potential pipeline for the implementation of augmented reality tutoring experiences through the use of a framework, such as GIFT, and a commercial see-through device, the HoloLens.

In section two of this paper we go through related work in the field. Section three goes into the interface integration with GIFT along with a detailed description of the setup which lead to GARAT. Section four illustrates the detailed usage of our authoring tool along with the creation of the two above training courses. Section five wraps up our research with a discussion of our conclusions and a recommendation for future work.

RELATED WORK

Augmented Reality Scenarios

Lee's SpaceTop (Lee, Olwal, Ishii, & Boulanger, 2013) and Xin's Napkin Sketch (Xin, Sharlin, & Sousa, 2008) provide guidelines for the seamless transition between 2D and 3D interaction; the former is presented as a single hybrid workspace, while the latter is a Mixed-Reality space interacting with a sketch. Pilot studies from both works indicate high user satisfaction with both interfaces, where users found the system quite intuitive and easy to grasp. A participant from SpaceTop, however, experienced arm fatigue during the trials, and overall, SpaceTop is not sufficiently responsive and accurate. Napkin Sketch was found to foster creativity, but users found the system laborious when asked to calibrate their best perspective for sketching. These considerations are important when working with interfaces involving physical load in the case of sketching on 3D. To address these issues our prototype has a real time update and an in-site 2D interface of the scenario being authored.

Authoring AR & Tutoring Systems

Several commercial and academic tools have been proposed to author AR scenarios. (Mota, Roberto, & Teichrieb, 2015) proposes a classification of the different tools and concludes with a taxonomy composed by two authoring paradigms: stand-alone and AR-plugins and two distribution strategies: platform-specific and platform independent. Our prototype falls under the category of AR-plugins because it is an add-on for GIFT and platform independent since GIFT offers capabilities to integrate with any external application. Authoring in immersive scenarios has been widely studied in (Lee, Nelles, Billinghurst, & Kim, 2004) where they set specific guidelines for authoring tasks and objects behaviors in tangible AR applications. Different domains have benefitted from AR combined with ITS. For example in the military domain, soldiers require training on physical tasks but the adaptive training techniques are being instructed in a desktop environment (Goldberg B. , Sottilare, Brawner, & Holden, 2012). In order to extend ITS beyond the desktop, a pipeline for the development of a possible training scenario is introduced in (LaViola, et al., 2015). Additionally, an analysis of different smart glasses for tutoring in the wild is presented in (Sottilare & LaViola, 2015). Different factors and tradeoffs are exposed under different duties such as land navigation, maintenance, tactical planning, etc. The benefits of ITS combined with AR is demonstrated in (Westerfield, 2015), where two groups were trained in a motherboard assembly task. The group that was trained with AR without adaptive instruction underperformed in test scores compared to the group with intelligent support.

INTEGRATION WITH GIFT

GIFT uses service-oriented architecture (SOA) plugins which can be developed to allow external applications to communicate with the framework. In Figure 2, a reduced version of the GIFT functional blocks displays our modified components in red. We implemented a new interop to interface with Unity 3D and a XML Remote Procedure Call (XMLRPC) server and client to communicate GIFT with Windows Universal Platform Applications (HoloLens). The plugin is used when authoring a course in the GIFT Authoring Tool (GAT) and also when a course is started from the Tutor User Interface (TUI). The training application connects via XMLRPC in a seamless exchange of information based on the course assessment definitions just like it does in the examples provided by GIFT.

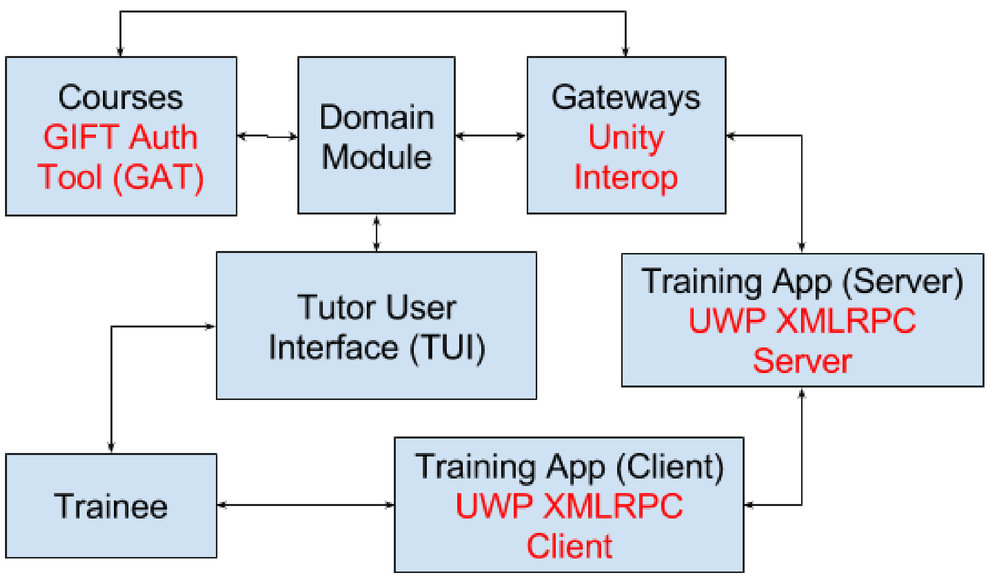


Figure 2. Modifications made to the GIFT functional blocks. Red text denotes additions to expand functionality. Note the new Unity interop option added in the GIFT Authoring Tool as well as an XMLRPC implementation for Universal Windows Platform.

Authoring an AR Scene in the GAT

For the authoring of an Augmented Reality Scenario we introduce a plugin we call the GIFT Augmented Reality Authoring Tool (GARAT) which is an interface to visualize and interact with 3D information. To access the tool an external application from the course objects should be added to the course flow. Inside its attributes are the application type: Unity application will invoke the GARAT plugin. The tool is divided in three sections left, center and right panels. The left panel shows assets located in the GIFT user directory, its properties and a preview of it. The center panel presents a 3D canvas initially empty which can contain information imported from assets or an external third party application e.g. room mesh. It also provides buttons to apply transformations to the assets, manage the 3D mesh acquisition server, load previously saved mesh, save mesh from the 3D canvas, clear the 3D canvas and generate a color point cloud. The 3D data in the canvas can be visualized as Mesh, Point Cloud and Wireframe via radio buttons on the bottom of the canvas, as shown in Figure 3.

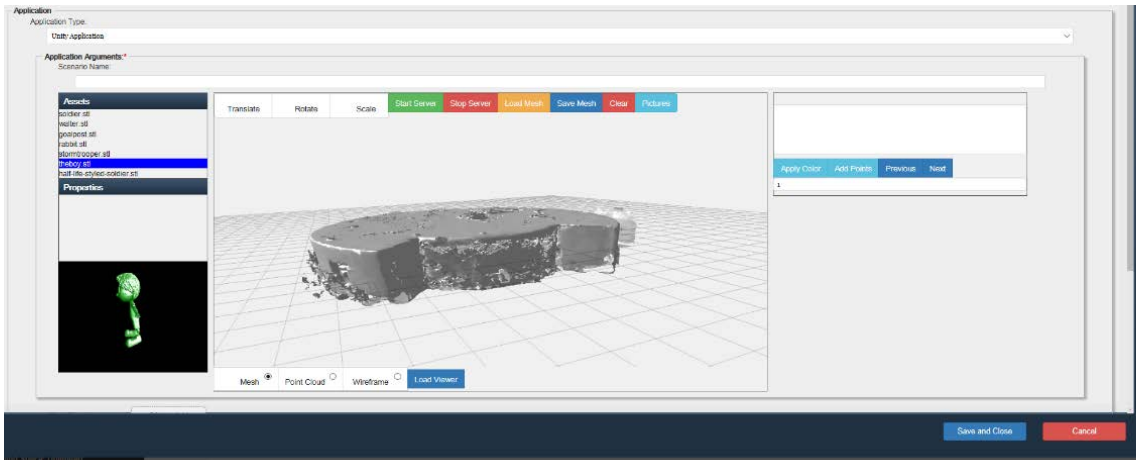


Figure 3. GARAT user interface. The left panel shows a preview of an asset. The central panel shows the mesh of a room as well as the different buttons to interact with.

3D Reconstruction from HoloLens

An important aspect in the authoring of an AR scene is to provide the author a spatial mapping of the physical space. To achieve this a universal windows application for HoloLens is developed. Once installed on the HoloLens, the authoring application can be started from the start menu. There are two modes of interaction: by voice and by gaze-tap. In order to start retrieving the geometry of the room, first from GARAT the plugin needs to be listening by clicking the “Start Server” button. If successful the tool will be ready to receive the data. Next, from the HoloLens application, a 3D map of the environment can be acquired by verbally saying “Start Scan” or “Open Menu”. The “Open Menu” statement will place the menu and allow the user to gaze-tap on the “Scan” button. A wireframe render around the geometry of the physical elements in the room is shown as can be seen in Figure 4. The information being rendered can be sent to GIFT to be displayed in GARAT by saying “Send Mesh” or by saying “Open Menu” followed by a gaze-tap on the “Send” button. It is important as well to attach anchors in the scene by verbally saying “Open Menu” followed by a gaze-tap on “Set Anchors”, then proceed to gaze at an specific point in the room and tap to place the anchor; this is for aligning HoloLens and GARAT 3D canvas coordinate systems when the tutoring application starts.

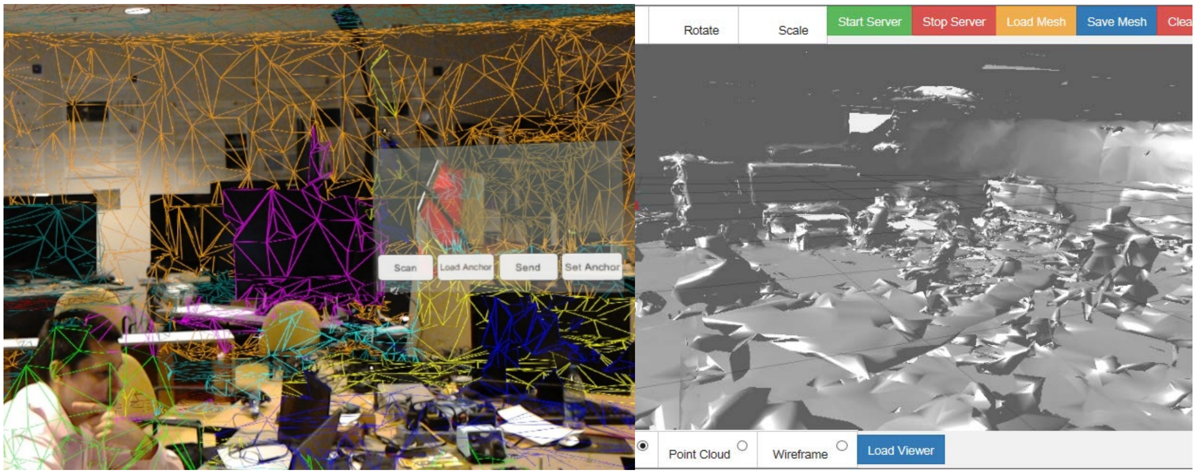


Figure 4. Left image shows the HoloLens application to scan a room with the menu controls open. Right image shows the mesh representation of the scan visualized in the GARAT 3D canvas.

The mesh generated from the reconstruction does not provide color information. In order to mitigate this problem a point cloud is generated from the mesh using RGB information coupled with extrinsic and intrinsic parameters retrieved from the HoloLens front camera. According to Windows Holographics documentation the extrinsic parameters store the camera's pose in the camera coordinate system, on the other hand the intrinsic properties (focal length, center of projection, skew) represent the projection transform mapped onto an image plane that goes from -1 to +1. This information is captured by saying "Take Picture" which will pack the data and upload it to the GIFT server. In the GARAT interface the pictures can be visualized on the right panel and can be traversed by clicking "Next" and "Previous" buttons. The point cloud is generated by first clicking on "Pictures", then selecting the image and finally clicking "Apply Color". As can be seen in Figure 5, a color point cloud was generated from the picture information in the right panel. One disadvantage of this method is that it needs to be repeated by each picture as the HoloLens interface does not allow complete control over the 3D information acquired (RGB, depth and camera parameters) in real time.

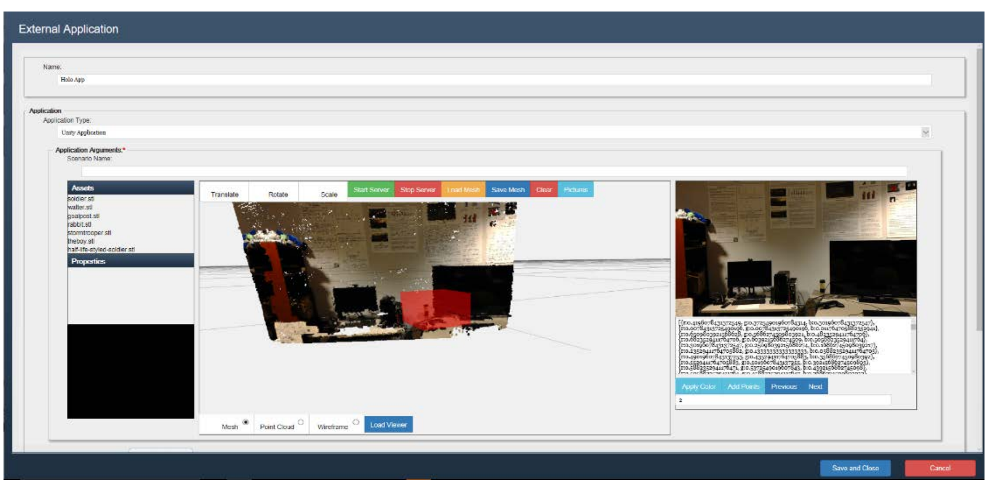


Figure 5. Point Cloud generated from the information in the right panel. The red cube represents the position and orientation of the camera the picture was taken with.

AUTHORING AUGMENTED REALITY ITS

The following section involves content to be learned by a student. Our study focuses on the external application course object that corresponds to the user interface proposed in this paper to author AR scenarios. For the first tutoring system, once a 3D room is loaded in the canvas, from the HoloLens application, anchors should be loaded to align coordinate systems between HoloLens and the GARAT 3D canvas. Next, in the left panel of the GARAT assets can be visualized and added to the scene via drag and drop as can be seen in Figure 6. The camera can be moved via pan, zoom and tilt with mouse or touch interactions. An object can be selected by clicking on it and transformations can be applied to it or information can be set in the properties panel, such as the object’s name. In the HoloLens the author can visualize the 3D elements in the GIFT interface as well as in the real world. The course can then be saved and the assets in the HoloLens tutoring application will be loaded based on the name and the transformations applied in GARAT.

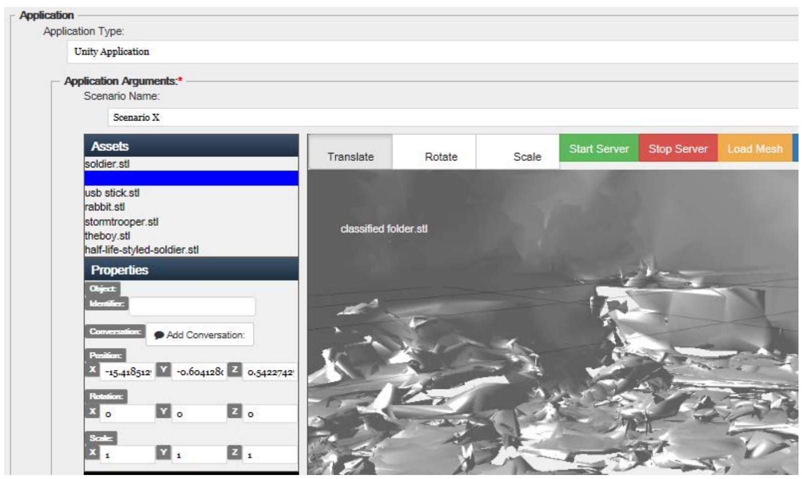


Figure 6. An asset being dragged to the scene, the label represents the 3D object.

The communication between GIFT and the tutoring application is defined in a real time assessment file as known as (DKF) which is appended to the external application. This file maps actions in the tutoring application triggered by messages to instructional strategies in GIFT. Since our purpose is to introduce GARAT, no additional assessment conditions were developed, instead, an existing condition assessment from GIFT was selected. Assessments were performed at the concept level via the StringMatchingCondition which defines concepts based on <key,value> sets, which trigger feedback responses based on the external application input as shown in Figure 7. The same configuration is used for both of the examples proposed in this work.

To summarize, we can define the process of Authoring in the following steps:

- Create a Unity external application object.
- From the GARAT interface, click start server which runs an XMLRPC server and client instance on GIFT.
- Start Authoring Application on HoloLens. (Acquire 3d reconstruction, place anchors, etc.)
- Visualize the scene on GARAT, place objects and label them.
- Append a Real Time Assessment.

- Save the course

We have extended the process of tutoring:

- In the HoloLens external application transforms are applied to game objects defined in GARAT.
- Via messages sent to GIFT which need to be defined via XMLRPC depending on users input.

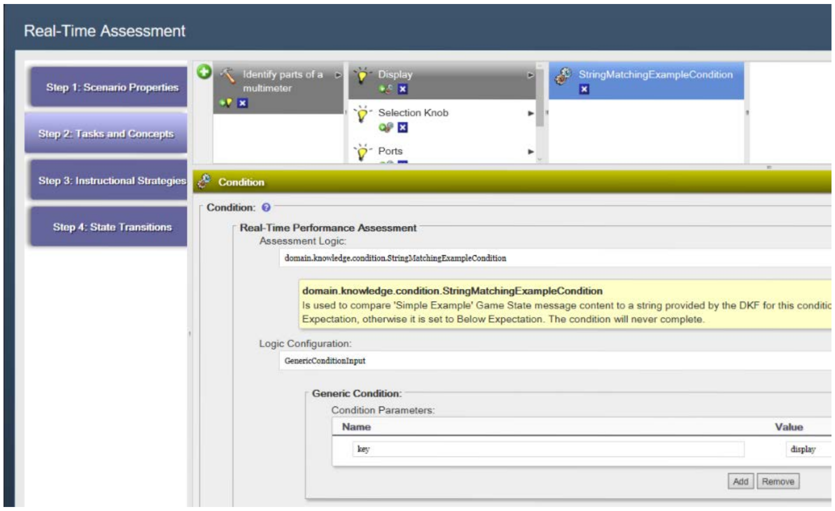


Figure 7. Real Time Assessment file for defining multimeter parts concepts.

Tutoring System to identify security leaks on a SCIF

The objective of this tutoring course is to brief a user on specific rules for handling classified information. As shown in Figure 8, a classified folder is left unattended on top of the desk and an unlabeled USB stick is left within the computer. For this specific problem, in pursuance of authoring an AR course, an awareness of the physical dimensions of the office is required. After a 3D scan has been performed on the room and sent to GARAT we proceed to place 3D objects accordingly in positions we expect the trainee to visualize them, an example of this placement is shown in Figure 8. The 3D objects are mapped to game objects in the HoloLens client application, however interaction events that will send messages to the GIFT server depend upon the client application specifics. The user is asked to “Identify risk threats in the office”. Gaze-tap on top of the elements is used to let GIFT know risk threats are being identified. The messages are mapped to a real time assessment file defined for the course through a StringMatchingCondition class. The feedback on HoloLens is displayed in a modal pop up window overlaid on screen space. The course ends when all risk threads have been identified.

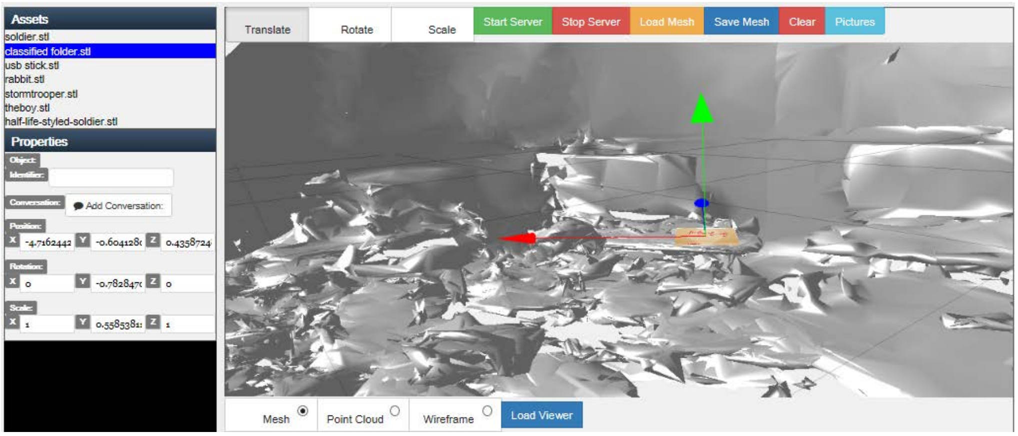


Figure 8. Applying transformations to the classified folder added to the scene mesh. Properties of the object are set on the left panel.

Tutoring System on the use of a Multimeter

Augmented Reality not only takes advantage of the spatial information (room geometry) but also can be used for object recognition. In this matter, we developed a course that augments information over a device, in this case a multimeter, to achieve some level of training. This example requires more development on the client due to the object recognition. Vuforia library allows to obtain a 3D mapping of objects with dimensions around 15 x 15 x 5 cm. This information can be fed into a HoloLens application and the object becomes trackable in HoloLens space. Messages are sent via the Unity interop to request information accordingly to the area of the device being gaze-tapped. In this way if the user gaze and tap on the selection knob, GIFT provides information about this part of the multimeter as can be seen in Figure 9. The assessment performance is defined in a Real Time Assessment file (see Figure 7). The information displayed over the tracked object is received as feedback from GIFT when the user gaze and tap on the corresponding part of the multimeter. The gaze and tap event sends a message back to GIFT with a key value indicating the instructional strategy to be triggered based on this input.

Figure 9. By gazing and tapping at the display, information displayed in the TUI is received in the HoloLens external application.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

There are many challenges that arise when developing training applications but the work can be simplified by extending the capabilities of existing tools. The development of a complete tutoring experience from GIFT represents a challenge due to the different capabilities, interaction techniques, physical tasks, etc that could be needed. However, in this work we aim to ease the pipeline introducing the GARAT tool. Leveraging most of GIFT capabilities directly on the interface pose a challenge, having a separated interface to define concept, tasks and assessment conditions could lead to jump back and forth between GARAT and the modal window. In the case of the device tutoring course a more natural interaction with the equipment is required e.g. identifying that the selection knob is placed on milliamps or voltage or selecting if the correct wire is connected to the right port. As done with the spatial awareness in GARAT an authoring tool can be implemented to ease this process.

This contribution of this work can be summarized as:

- An initial exploration in the use of GIFT as a generator of augmented reality tutoring courses.
- A tool for content generation of Augmented Reality scenarios that can be scaled to produce more complex behaviors and training scenarios inside GIFT.
- An XMLRPC server[3] implementation to connect Windows Universal Apps to GIFT.

The ability to connect to external platforms open the possibilities to use current or future head mounted displays that can provide better user experience. The architecture is built upon a modular architecture which can be reused across different external application platforms. HoloLens is a compact hardware which allows to have a rough 3D mapping of the world (coarse mesh), however, its internal algorithms are closed source code and it provides limited data, specifically RGB and depth information in real time. The tutoring experiences presented in this work were constrained to the HoloLens predefined interaction techniques. The GIFT user interface in some cases tend to hide some capabilities with the overuse of modal windows. Adding an interop requires some steps that can possibly be simplified. In future research, a system that requires minimum programming needs to be developed and tested with regular users.

The interface we have developed is a prototype that provides the first step toward the creation of advanced authoring tools which can easily merge intelligent tutoring systems with augmented reality. While shortcomings exist in the hardware and software chosen, we demonstrated potential work arounds or future solutions which will improve the system further. With each iteration developed, authoring tools such as GARAT will simplify and solve the obstacles faced in the creation of augmented reality training courses.

REFERENCES

Goldberg, B., Sottilare, R., Brawner, K., & Holden, H. (2012). Adaptive game-based tutoring: Mechanisms for real-time feedback and adaptation. *International Defense \& Homeland Security Simulation Workshop in Proceedings of the I3M Conference.*

Goldberg, B., Sottilare, R., Moss, J., & Sinatra, A. (2016). Dimensions of Instructional Management Research in Support of the Generalized Intelligent Framework for Tutoring (GIFT). *International Defense \& Homeland Security Simulation Workshop of the I3M Conference.*

Goodwin, G., Johnston, J., Sottilare, R., Brawner, K., Sinatra, A., & Graesser, A. (2015). *Individual learner and team modeling for adaptive training and education in support of the US Army Learning Model: Research*

[3] https://github.com/andnovar/xmlrpc-universal/tree/xmlrpc_server

Outline. Tech. rep., ARMY RESEARCH LAB ABERDEEN PROVING GROUND MD HUMAN RESEARCH AND ENGINEERING DIRECTORATE.

LaViola, J., Williamson, B., Brooks, C., Veazanchin, S., Sottilare, R., & Garrity, P. (2015). Using Augmented Reality to Tutor Military Tasks in the Wild. *Interservice/Industry Training, Simulation, and Education Conference 2015.*

Lee, G. A., Nelles, C., Billinghurst, M., & Kim, G. J. (2004). Immersive authoring of tangible augmented reality applications. *Proceedings of the 3rd IEEE/ACM international Symposium on Mixed and Augmented Reality*, (pp. 172-181).

Lee, J., Olwal, A., Ishii, H., & Boulanger, C. (2013). SpaceTop: Integrating 2D and Spatial 3D Interactions in a See-through Desktop Environment. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 189-192). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2470654.2470680

Mota, R. C., Roberto, R. A., & Teichrieb, V. (2015). [POSTER] Authoring Tools in Augmented Reality: An Analysis and Classification of Content Design Tools. *Mixed and Augmented Reality (ISMAR), 2015 IEEE International Symposium on*, (pp. 164-167).

Renambot, L., Marrinan, T., Aurisano, J., Nishimoto, A., Mateevitsi, V., Bharadwaj, K., . . . Leigh, J. (2014, 4). SAGE2: A collaboration portal for scalable resolution displays. *Future Generation Computer Systems, 54*, 296-305.

Sottilare, R. A. (2015). Dimensions and Challenges in Domain Modeling for Adaptive Training. *Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym3)*, (p. 103).

Sottilare, R. A., Brawner, K. W., Goldberg, B. S., & Holden, H. K. (2012). The generalized intelligent framework for tutoring (GIFT). *Orlando, FL: US Army Research Laboratory--Human Research \& Engineering Directorate (ARL-HRED).*

Sottilare, R., & LaViola, J. (2015). Extending Intelligent Tutoring Beyond the Desktop to the Psychomotor Domain. *Interservice/Industry Training, Simulation, and Education Conference 2015.*

Sottilare, R., Ososky, S., & Boyce, M. (2015). Domain Modeling in Support of Adaptive Instructional Decisions in the Generalized Intelligent Framework for Tutoring. *International Defense \& Homeland Security Simulation Workshop of the I3M Conference.*

Statt, N. (2015). Microsofts HoloLens explained: How it works and why its different. *Microsofts HoloLens explained: How it works and why its different*. R{\'e}cup{\'e}r{\'e} sur CNET: http://www. cnet. com/news/microsoft-hololens-explained-how-itworks-and-why-its-different.

Westerfield, G. a. (2015). Intelligent Augmented Reality Training for Motherboard Assembly. *International Journal of Artificial Intelligence in Education*, *25*(1), 157-172. Retrieved from http://dx.doi.org/10.1007/s40593-014-0032-x

Xin, M., Sharlin, E., & Sousa, M. C. (2008). Napkin Sketch: Handheld Mixed Reality 3D Sketching. *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology* (pp. 223-226). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/1450579.1450627

ABOUT THE AUTHORS

Andrés N. Vargas González *is a PhD student at the University of Central Florida in the department of Electrical Engineering and Computer Science within the Interactive Systems and User Experience Research Cluster of Excellence under Dr. LaViola. His primary research includes pen-based computing and 3D user interfaces. While in Ecuador he won a Fulbright fellowship from 2012 to 2014 leading to the completion of his Masters degree under Dr. LaViola with a thesis on sketch based interaction for charts creation and manipulation. In 2015 he was selected by the Kivy project under the Python Software Foundation to participate in a Google Summer of Code program.*

Brian M. Williamson *is a faculty researcher at the University of Central Florida in the department of Electrical Engineering and Computer Science within the Interactive Systems and User Experience Research Cluster of Excellence under Dr. LaViola. His primary research includes 3D user interfaces with previous papers written on the RealNav system, a natural locomotion system evaluated for video game interfaces. At the University of Central Florida, Brian has published a thesis on natural locomotion, and has been published by IEEE and CHI.*

Dr. Joseph J. LaViola Jr. *is the CAE Link Professor and associate professor in the Department of Electrical Engineering and Computer Science and directs the Interactive Systems and User Experience Research Cluster of Excellence at the University of Central Florida. He is also an adjunct assistant research professor in the Computer Science Department at Brown University. His primary research interests include pen-based interactive computing, 3D spatial interfaces for video games, human-robot interaction, multimodal interaction in virtual environments, and user interface evaluation. His work has appeared in journals such as ACM TOCHI, IEEE PAMI, Presence, and IEEE Computer Graphics & Applications, and he has presented research at conferences including ACM SIGGRAPH, ACM CHI, the ACM Symposium on Interactive 3D Graphics, IEEE Virtual Reality, and Eurographics Virtual Environments. He has also co-authored "3D User Interfaces: Theory and Practice," the first comprehensive book on 3D user interfaces. In 2009, he won an NSF Career Award to conduct research on mathematical sketching. Joseph received a Sc.M. in Computer Science in 2000, a Sc.M. in Applied Mathematics in 2001, and a Ph.D. in Computer Science in 2005 from Brown University.*

Dr. Robert A. Sottilare *leads adaptive training research within the US Army Research Laboratory where the focus of his research is automated authoring, automated instructional management, and evaluation tools and methods for intelligent tutoring systems. His work is widely published and includes articles in the Cognitive Technology Journal, the Educational Technology Journal, and the Journal for Defense Modeling & Simulation. Dr. Sottilare is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT), an open-source tutoring architecture, and he is the chief editor for the Design Recommendations for Intelligent Tutoring Systems book series. He is a visiting scientist and lecturer at the United States Military Academy and a graduate faculty scholar at the University of Central Florida. Dr. Sottilare received his doctorate in Modeling & Simulation from the University of Central Florida with a focus in intelligent systems. In 2012, he was honored as the inaugural recipient of the U.S. Army Research Development & Engineering Command's Modeling & Simulation Lifetime Achievement Award.*

Science is Zarked: An Intelligent Tutoring System for Learning Research Methods

Samantha F. Warta
University of Central Florida Institute for Simulation and Training
swarta@stetson.edu

INTRODUCTION

Intelligent tutoring systems (ITSs) have two primary goals: a) specifying *what* concepts to teach an individual learner and b) *how* to teach them through personalized instructional strategies (Ohlsson, 1987; Wenger, 1987). Science is Zarked is an ITS, designed and validated with the Generalized Intelligent Framework for Tutoring system (GIFT; Sottilare, Brawner, Goldberg, & Holden, 2012), for teaching a basic course on research methods. The motivation for constructing this ITS stems from the need to address a low-level training gap within university-associated labs. That is, no accredited system intended to train and evaluate students on their level of research methods knowledge currently exists. Specifically, this ITS is targeted at students applying for positions as undergraduate research assistants in university-associated labs. As such, redressing this training gap is essential and through the creation of Science is Zarked, university-associated labs will save both time and money.

While teaching aspects of the scientific method and various research techniques applicable to most scientific disciplines, this tutor aims to use other ITSs dedicated to science education as the foundation for designing a system targeted at learners with very little knowledge about science or research methods. To that end, Science is Zarked is grounded in a pedagogy-oriented approach to aid exploration of the programmatic learning content structure of this ITS. That is, this ITS focuses on the sequence of the material taught and the strategies used to teach the content. Specifically, this system employs pedagogical strategies such as: (a) an adaptive courseflow, to adjust to individual learner characteristics – such as, existing knowledge, desire for feedback, and performance – in an effort to positively influence learning outcomes; (b) a programmatic content structure, which emphasizes the retention of concepts through gradual introduction and repetition to enable learners to develop a genuine understanding of scientific research; and (c) the case method of instruction, to bridge the gap between theory and application. In addition to gaining greater understanding of how these teaching strategies influence learning outcomes, the primary goal of this ITS is to encourage learners' interest in the sciences and demonstrate the ease of mastery of relatively basic scientific concepts.

As such, the first section of this paper will examine literature related to GIFT and ITSs for science education. The second section will introduce Science is Zarked, the ITS central to this paper, and further describe the problem it was intended to address. Additionally, this section will review design decisions and provide an in-depth examination of the system's structural components as well as detail the pedagogical strategies employed. As a final point, this paper concludes with a review of the lessons learned, recommendations for GIFT features to provide further functionality in this domain, and future plans for Science is Zarked.

GIFT

GIFT, the Generalized Intelligent Framework for Tutoring (Sottilare et al., 2012), is a system intended to aid in the design and generation of computer-based tutoring systems. Developed under the Adaptive Tutoring Research Science & Technology project, GIFT represents a system grounded in empirical study. The development of this framework of tools is supported by researchers at the Learning in Intelligent

Tutoring Environments (LITE) Laboratory, part of the U.S. Army Research Laboratory – Human Research and Engineering Directorate (ARL-HRED) and was designed to facilitate study of computer-based tutoring systems throughout the government, industry, and academia (ARL GIFT, 2016).

Relevant to this discussion is Murray's (1999) classification of ITS authoring tools, as either pedagogy or performance-oriented, in accordance with each system's capabilities (see Figure 1 for a visual representation of this conceptual mapping). Pedagogy-oriented systems primarily emphasize the teaching and sequencing of content. On the other hand, performance-oriented systems are most concerned about learning outcomes, focusing on teaching learners by allowing them to practice learned skills while receiving feedback. However, GIFT does not fall into only one of these categories as it enables the design of an ITS with pedagogy and/or performance-oriented features. Accordingly, four modules characterize GIFT's capabilities: the sensor, learner, pedagogical, and domain modules. The sensor module enables the monitoring of individuals through commercial sensors and provides an interface to GIFT while formatting, processing, and storing the collected data. The domain module is concerned with providing domain-specific content, assessment, and feedback. However, the domain module only provides feedback when the pedagogical module determines it is necessary. Lastly, the learner module assesses an individual's cognitive and affective state through the tracking of performance, historical, and sensor data.

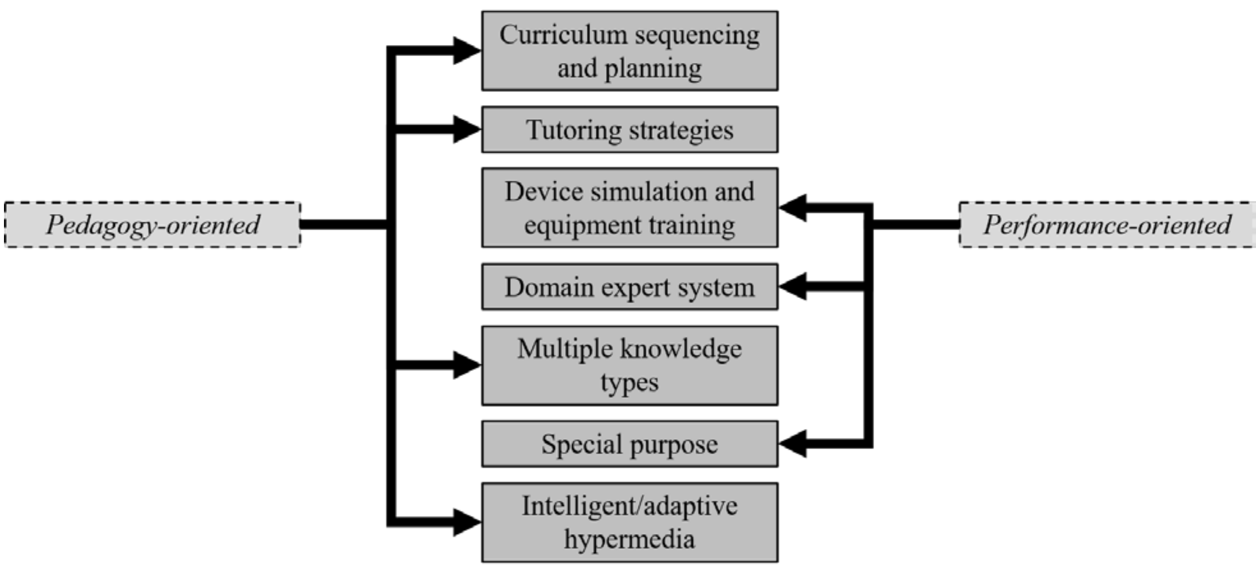


Figure 1. Conceptual Mapping of Murray's (1999) ITS Capability Classifications.

Intelligent Tutoring Systems for Science Education

A brief review of the literature yielded three ITSs designed specifically for varying facets of scientific education. These tutors consist of My Science Tutor (Ward et al., 2013), the Modeling and Inquiry Learning Application (Joyner & Goel, 2015), and the Andes system (Vanlehn et al., 2005). All three systems provide valuable insight into the teaching of science-related concepts and a useful comparison to Science is Zarked, as well as inspiration to guide the development of future GIFT capabilities to support ITSs in this domain.

My Science Tutor (MyST; Ward et al., 2013) is an ITS that teaches scientific concepts targeting elementary school students. Using an avatar named Marni, MyST is designed to employ conversational dialogues supplemented by illustrations, animations, and interactive simulations to teach learners scientific concepts. Ward et al. (2013) split learners into three groups: a classroom of learners taught under normal conditions by a single teacher to serve as a control, one on one tutoring with a human tutor, and one on one tutoring with a virtual tutor. Results indicated that both of the one on one tutoring groups had significantly higher learning gains than the control groups. However, there was no significant difference in learning gains between the two tutoring groups. Thus, evidence supported the assertion that expert human tutoring is equivalent to virtual tutoring in the case of MyST.

The Modeling and Inquiry Learning Application (MILA; Joyner & Goel, 2015) represents another example of a system designed for science education. MILA is a metacognitive tutoring system that uses inquiry-driven modeling to teach various scientific concepts. Using this system, learners create a model to describe a phenomenon while MILA-Tutoring (MILA-T), an intelligent agent, monitors and responds to learners' behavior. MILA-T represents a pedagogical agent, divided into five different versions: a Guide, a Critic, a Mentor, an Interviewer, and an Observer. The Guide and the Critic do not provide feedback unless asked whereas the Mentor, Interviewer, and Observer interrupt a learner's actions when appropriate. However, the Observer primarily operates in the background by feeding information to the other agents. Joyner and Goel (2015) used five classes of learners, placing two into a control group and three into an experimental group. The control group utilized MILA while the experimental group used MILA with the addition of MILA-T. Comparing interaction logs between the control and experimental groups, Joyner and Goel examined how MILA-T influenced learners' modeling and inquiry processes. They provided evidence to support the assertion that learners' engagement was greater with the MILA-T addition. Specifically, their results suggested that learners utilized the feedback and retained the information received from the tutoring system. Learners were most likely to revise their models and expound upon them after tutor interactions. Thus, MILA-T not only improved learners' engagement by positively influencing their disposition, but also improved their performance on modeling and inquiry tasks.

The Andes system (Vanlehn et al., 2005) is an intelligent physics tutoring system designed to improve learner performance through interaction. Studied at the United States Naval Academy, Andes significantly improved student learning. According to Vanlehn et al. (2005), the key to Andes' success was the form of answers elicited from learners, representing a "whole derivation, which may consist of many steps, such as drawing vectors, drawing coordinate systems, defining variables and writing equations" (p. 147). Here, the focus is not on the content, but rather the method with which it is taught.

SCIENCE IS ZARKED

Science is Zarked is an ITS designed to teach a basic course on research methods through the GIFT authoring system (https://cloud.gifttutoring.org/). In addition to teaching aspects of the scientific method and various research techniques, this tutor also aims to provide the best practices related to each research method. While the content within this tutor was not meant to be representative of an introductory level research methods course that spans an entire school semester, it offers enough content to provide a broad perspective relating to several different experimental designs and research methods utilized in various scientific disciplines. The current version of Science is Zarked can be accessed at the following URL: https://cloud.gifttutoring.org/tutor/?eid=a4b87263-e3bd-47d3-a0be-b1bd0fda3980.

Rationale and Benefits

Employees in university-associated labs regularly hire undergraduate students to assist with various tasks on projects, like experimental design and data collection. Usually the students hired have little to no experience in a scientific research setting and arrive with nothing more than the knowledge retained from basic high school science courses. The problem here is that to successfully and effectively collect data during an experiment, some knowledge of scientific research methods, beyond what students learned in high school, is required. However, there is currently no accredited system that ensures students have at least a basic understanding of research methods, so lab employees must train each individual student they hire every semester. This has the potential to be both a time consuming and labor-intensive process, depending on the individual student. Thus, the creation of an ITS designed to teach these basic research methods concepts will be beneficial to labs by enabling them to save time and money when training new undergraduate research assistants.

Design and Structure

As mentioned previously, specific pedagogical strategies utilized by Science is Zarked include: (a) an adaptive courseflow, (b) a programmatic content structure, and (c) the case method of instruction. Overall, this ITS seeks to provide a solid coverage of the basic concepts presented to learners during an introductory research methods course, offering supportive material and organizing the modules to enhance student learning outcomes. This ITS's design emphasizes the retention of research concepts through gradual introduction to terms and demonstration of their application through case studies. In particular, this enables learners to develop an understanding of scientific research as an interconnected and integrated process of thinking rather than a series of disembodied concepts. While merely an introduction to various research methods and experimental approaches, this ITS underscores the importance of empirical research, and the methods detailed within, to build upon current scientific knowledge. See Figure 2 below for an overview of the structure of the tutor.

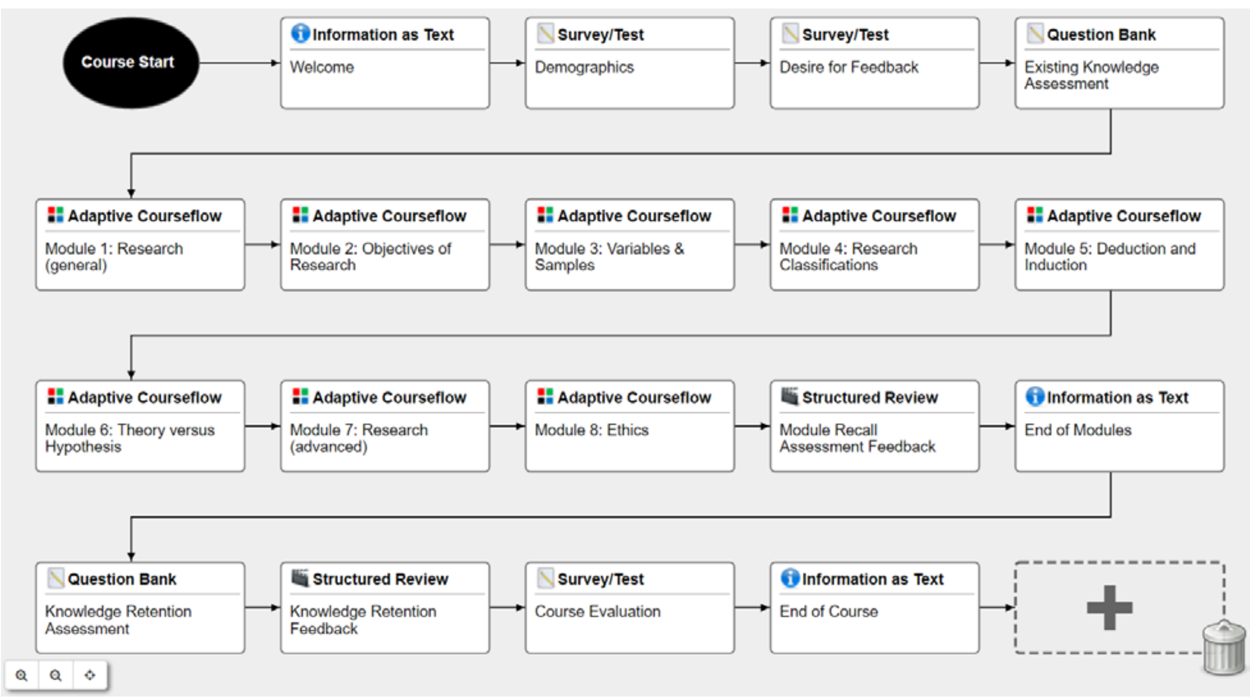


Figure 2. Structure of the Science is Zarked Intelligent Tutoring System.

Course Concepts

Comprised of eight different modules, Science is Zarked is structured such that a student with very little knowledge about science or research methods within any scientific discipline is capable of comprehending and gaining applicable knowledge as well as insight from the course materials. Likewise, a more experienced student, that may have taken a course or two in research methods, also shares the possibility of learning something new. The general layout of this tutoring system begins with a module that provides an introduction to scientific research and empirical research methods. This first module teaches the learner what research is and what it is not by providing material on several relevant concepts and terms. Additionally, this module outlines the scientific method and makes an effort toward outlining the qualities of good research. The modules that follow build upon one another and become increasingly more complex, covering topics ranging from the objectives of research, deductive and inductive reasoning, the difference between hypotheses and theories, comparing the various systems of research; to addressing more advanced topics like ethics in research as well as characteristics of samples and variables, reliability, and validity. See Table 1 for a full listing of the modules and each concept they cover. Items for the recall assessments in these modules were generated based upon previously taken research methods courses, reviewing the literature to attain content validity, and modified from several web-based sources (Dattalo, n.d.; Marley, 2007). Future updates to Science is Zarked, a GIFT course export file, and the material presented in the learning phase and recall assessment for each module can be downloaded from https://www.researchgate.net/profile/Samantha_Warta.

Table 1. Science is Zarked Modules and Concepts Covered.

Module	Name	Concepts Covered
1	Research (general)	Basic concepts and terms, what is research?, qualities of good research, the scientific method
2	Objectives of Research	Descriptive, correlational, explanatory, and exploratory research
3	Variables and Samples	Independent and dependent variables; the importance of operationalization; continuous, non-continuous, and extraneous variables; samples; random and non-random sampling; characteristics of good samples; sample biases
4	Research Classifications	Basic and applied research, quantitative and qualitative research, experimental and non-experimental research
5	Deduction and Induction	Deductive and inductive reasoning
6	Theory versus Hypothesis	Hypotheses, characteristics of a good hypothesis, theories, characteristics of a good theory
7	Research (advanced)	Rule of parsimony, replication, reliability, and validity
8	Ethics	Ethics in research, use of the institutional review board, informed consent, harm and risk, deception

Pedagogical Features

Adaptive Courseflow

ITSs that are adaptive in nature offer learners a uniquely tailored learning experience based on the individual needs of a particular student. Three learner attributes that are important for a tutoring system to take into account are learner knowledge (Gertner & VanLehn, 2000), learner cognitive skills (Arroyo et al., 2004; Royer et al., 1999), and learner attitudes (Arroyo & Woolf, 2005). Assessing existing learner knowledge provides students with an ITS that both benefits learning outcomes and is capable of a wide range of teaching techniques, while discerning the optimal teaching intervention (Gertner & VanLehn, 2000). Previous research has shown that pre-existing knowledge influences performance within ITSs (Taub et al., 2014; Trevors, Duffy, & Azevedo, 2014). Accordingly, Science is Zarked begins by assessing existing learner knowledge through administration of a pre-test on the concepts covered by all course modules. The primary objective of this pre-test is to feed data, pertaining to each individual learner, into the ITS to establish their existing knowledge base and influence the course flow. As GIFT capabilities permit, this will mean providing learners with the option to skip the learning phases within modules in which they have demonstrated mastery of a given concept by correctly answering all question items associated with that module during the pre-test.

Collecting information pertaining to a learner’s cognitive skills and adapting a tutor based on this has the added benefit of improving learning outcomes (Arroyo et al., 2004; Royer et al., 1999). Since cognitive skills can be reasonably characterized by processes governing thinking, attention, learning, memory, and reasoning, then a simple measure of an individual’s cognitive skills can be gathered by inquiring about the highest level of education achieved (Ceci, 1991). Within the demographics survey object, learners answer questions relating to not only their highest level of education achieved, but also their major or focus of study and their science course history. This establishes an additional measure that functions as another check on the existing knowledge assessment and support determination of level of expertise.

Following this, the ITS transitions into an assessment designed to measure an individual’s desire for feedback (Moore, Erichsen, & Warta, 2014). According to Renkl (2002) and Wood and Wood (1999), a connection exists between learners’ behavior, attitudes, and perception. For example, when students interacted with an ITS and it provided meaningful feedback, this positively influenced the learning outcome by affecting attitudes (Aleven et al., 2003; Arroyo & Woolf, 2005). Given the connection between perception, behavior, and attitudes, it is crucial that the learner’s willingness to receive feedback be measured such that an ITS adapts appropriately. As a result, the pedagogical module within GIFT was set to recognize when learners scored high or low on the desire for feedback measure and will, accordingly, offer hints and question-by-question feedback as learners complete the recall assessment in each adaptive courseflow module.

After these surveys, eight adaptive courseflow modules teach learners a series of scientific concepts. Within these modules, the ITS presents course content to learners before they answer questions on a recall task. If learners do not score sufficiently high enough (i.e., answer approximately 80% of the questions correctly), then they are unable to advance to the next module and must repeat the current module’s learning phase until they receive an acceptable recall score. However, while the intent is for learners to be able to skip the learning phases of these adaptive courseflow modules as a function of their scores on the existing knowledge assessment, they will still be required to complete the recall questions during the assessment phase. This serves as a secondary check on learners’ existing knowledge to ensure that they truly understand the concepts taught in each module and did not just happen to guess the correct answer on the existing knowledge assessment.

Once learners have completed all eight modules, they are able to access a structured review, which provides the learner with a summary of all the assessments taken throughout the course as well as any feedback offered. Next, learners complete a knowledge retention assessment in the form of a post-test of the existing knowledge question bank. Scores from the knowledge retention assessment can then be compared to the existing knowledge assessment scores in a pre-test/post-test analysis. This will measure learner improvement or decline and ensure the effectiveness of the learning phase content in teaching through the three primary pedagogical strategies employed. Lastly, learners complete a course evaluation designed to assess the functionality of the course content and identify where the course has the potential for improvement. The course evaluation consists of subjective self-report answers and provides a useful comparison to the more objective measurement of learner performance throughout the course. Items for this measure were modified from the Berkley Center for Teaching & Learning course evaluations question bank (UC Berkley, 2016). This course evaluation asks learners to rate the clarity of the course content presented as well as its usefulness in completing each module assessment. Additionally, the course evaluation will serve as a "manipulation check" of sorts in that it will not only assess the effectiveness of the feedback provided to learners, according to their score on the desire for feedback items, but also their satisfaction with the course. In particular, this will serve to reinforce the validity of the desire for feedback measure and its inclusion as an adaptive component for Science is Zarked.

Programmatic Content

The structure of the learning content within Science is Zarked is best characterized as programmatic in nature. That is, difficult or unfamiliar concepts are introduced within the beginning modules to the extent that it facilitates this introductory discussion. Later modules then reexamine these concepts to provide a fuller picture. As learners progressively work through each module, the addition of new concepts to those already introduced provides a more coherent model of the empirical research process. Beginning at a very basic level, this ITS allows learners to progressively master the ideas presented, working their way up to more complex and comprehensive concepts. This design provides a rational and comprehensible experience of a very basic set of research methods by forcing each module to build upon the previous module rather than presenting learners with multiple, individual and independent, disembodied ideas. While this structure may seem rather repetitive in nature, and indeed allows for an adaptive courseflow that enables learners to repeat content, such repetition is actually useful to building knowledge that is more easily accessible on recall (DeKeyser, 2007; Kuczaj, 1983; Larsen-Freeman, 2012; Rydland & Aukrust, 2005; Weir, 1962). As mentioned in the previous section, the success or failure of such repetition throughout the course is easily verified in a pre-test/post-test analysis using the existing knowledge and knowledge retention scores.

Further, another element of the programmatic content within this ITS includes the use of graphical visualizations and simulations alongside the written material found within the learning phase of each module. While MyST could be reasonably classified as a performance-oriented system, Science is Zarked represents a pedagogy-oriented system much like MILA and Andes. That is, this ITS was designed to adapt to individual learners and focus on the sequence of the material taught as well as the strategies used to teach the content. In particular, throughout the course modules, Science is Zarked utilizes several graphical representations to illustrate scientific concepts, mirroring MyST's use of this media (see Figure 3).

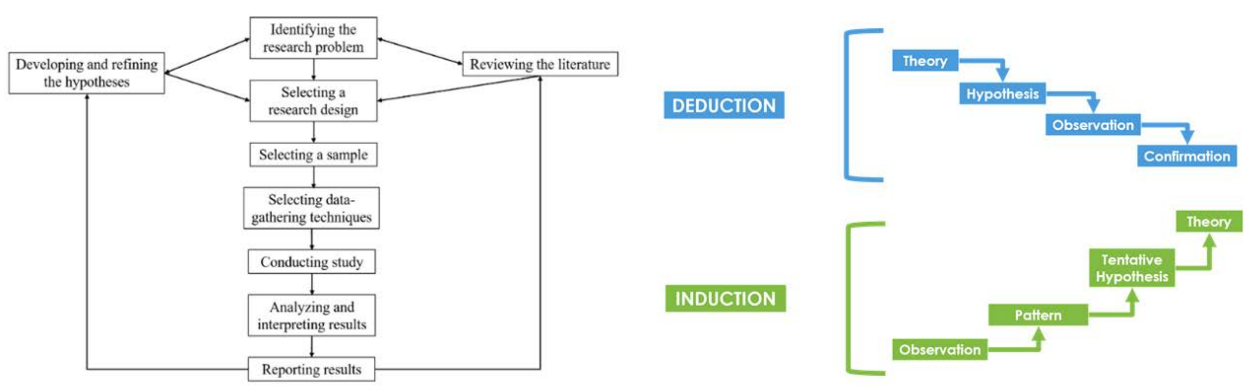


Figure 3. Visualizations used in Science is Zarked. Left*:* Steps of the Scientific Method (adapted from Blum & Blum, 2013). *Right:* Comparison of Deduction and Induction Approaches.

Then, following the module on *Variables and Samples in Scientific Research*, the ITS redirects learners to a natural selection simulation (Saul, 2005). This serves both as a fun activity and reinforces the content pertaining to the previous module. The simulation directs learners to manipulate the independent variable (mutation levels of the organisms) to see how it influences the sample. This type of simulation provides content similar to that found in MILA and the Andes system. Additionally, this type of content increases learner engagement with the material, utilizing the repetition and application of concepts to improve recall.

Case Method of Instruction

Similar to MyST, MILA, and the Andes system, one of the central teaching methods Science is Zarked utilizes is case studies (see Figure 4). In the lesson content within each module, short case studies help illustrate a particular concept, teaching application in addition to the strictly theoretical nature of the remaining lesson content. This strategy of teaching is crucial to developing learners' ability to apply theoretical knowledge to complex situations they may encounter. Also known as the case method of instruction, this approach was specifically designed to bridge the gap between theory and application (Jackson, 1985; Johnson & Purvis, 1987; Kleinfeld, 1990; Lee, 1983; Newey, 1987; Rasinski, 1989; Schwartz, Fiddes, & Dempster, 1987; Scully, 1984).

To test the hypothesis, "Listening to music lowers blood pressure levels", there are two ways of conducting research:

- **Experimental (helps determine causation):** Participants are divided into two groups. One group listens to music while the other does not. Compare blood pressure levels.
- **Correlational (does not determine causation):** Using a survey, ask participants how they feel in addition to how often they listen to music, and then compare the results.

Figure 4. Case Study used to illustrate the Differences between an Experimental and Correlational Design.

CONCLUSION AND RECOMMENDATIONS FOR FUTURE RESEARCH

The ITS development process in GIFT was not well-defined and challenging at times, since examples of successful systems built using GIFT were limited. During initial design of Science is Zarked, other scientific education ITSs acted as models to support the inclusion of validated teaching techniques and information elicitation methods. However, while GIFT excels in other areas, the options for eliciting information from learners and providing an interactive environment conducive to teaching scientific concepts were sparse. Approaches to collecting information from learners on surveys in GIFT were limited to multiple-choice questions, slider bars, and rating scales. Specifically, capabilities like those seen in MILA's interactive model-building interface or the ability to draw, define variables, or write equations like in the Andes system, would be useful in ITSs for science education. An interactive interface is requisite for modeling hypotheses and "running" a simulated experiment within the system to demonstrate different types of research methods or experimental effects, for instance. Alternatively, assessment items within an ITS could ask learners to arrange the procedural steps of the induction and deduction approaches, found in Figure 3, in the correct order. While these particular capabilities have been described herein to benefit an ITS focused on science education, they also have potential application in other domains as well.

Future plans for Science is Zarked include testing the efficacy of the ITS in its current form as well as adding an assessment of learners' attitudes at the mid-point of the course, so that it may adapt from that stage for the purpose of providing learners with a provisional and adjustable frequency of feedback. The addition of supplemental modules to those already covered are planned to include: a) specific experimental designs (e.g., between-subjects and within-subjects designs), b) experimental effects (e.g., practice effects, carry-over, etc.), and c) an introduction to basic experimental statistics. In effort to retain relevance and offer greater customization of this ITS to university-associated labs, additional modules could be added to cover special topics that are relevant to the existing projects for which a lab may currently be training undergraduate research assistants. Further planned improvements for this ITS also includes an addition of supplemental items to the module recall assessments for adaptive purposes, so that the difficulty level can be further tailored to each individual learner.

To conclude, Science is Zarked represents a novel contribution to the ITS community given that reviews of the literature did not reveal any other tutor, authored with GIFT or another system, addressing research methods. The purpose of this paper was to lay out the preliminary design of this ITS and importantly, demonstrate the applicability of GIFT as an essential tool in the design of ITSs focused on science education.

REFERENCES

Aleven, V., Koedinger, K. R., & Popescu, O. (2003). A tutorial dialog system to support self-explanation: Evaluation and open questions. In Proceedings of the 11th International Conference on Artificial Intelligence in Education (pp. 39-46). Amsterdam: IOS Press.

ARL GIFT. (2016). Generalized Intelligent Framework for Tutoring (GIFT; Version 2016-1) [Software, Web-based application]. Available from https://gifttutoring.org/projects/gift/wiki/Overview

Arroyo, I., Beal, C., Murray, T., Walles, R., & Woolf, B. P. (2004). Web-based intelligent multimedia tutoring for high stakes achievement tests. In Proceedings of the International Conference on Intelligent Tutoring Systems (pp. 468-477). Springer Berlin Heidelberg.

Arroyo, I., & Woolf, B. P. (2005). Inferring learning and attitudes from a Bayesian Network of log file data. In Proceedings of the 12th International Conference on Artificial Intelligence in Education (pp. 33-40). Amsterdam: IOS Press.

Blum, T. L., & Blum, A. J. (2013). A practical guide to doing research. DeLand, FL: Stetson University.

Ceci, S. J. (1991). How much does schooling influence general intelligence and its cognitive components? A reassessment of the evidence. Developmental Psychology, 27(5), 703.

Dattalo, P. (n.d.). Review questions and answers for foundations of social work research (SLWK 609) course. Retrieved from http://www.people.vcu.edu/~pdattalo/ReviewQsAns_609.htm

DeKeyser, R. (ed.). (2007). Practice in a second language. Perspectives from applied linguistics and cognitive psychology. Cambridge: Cambridge University Press.

Gertner, A. S., & VanLehn, K. (2000). Andes: A coached problem solving environment for physics. In International Conference on Intelligent Tutoring Systems (pp. 133-142). Springer Berlin Heidelberg.

Jackson, A. H. (1985). Teaching pediatric psychopharmacology: An interdisciplinary model. Journal of the American Academy of Child Psychiatry, 24, 103-108.

Johnson, J., & Purvis, J. (1987). Case studies: An alternative learning/teaching method in nursing. Journal of Nursing Education, 26, 118-120.

Joyner, D. A., & Goel, A. K. (2015). Improving inquiry-driven modeling in science education through interaction with intelligent tutoring agents. In Proceedings of the 20th International Conference on Intelligent User Interfaces (pp. 5-16). ACM.

Kleinfeld, J. (1990). The case method in teacher education: Alaskan models. ERIC Clearinghouse on Rural Education and Small Schools. Digest EDO-RC-90-6, 2-3.

Kuczaj, S. (1983). Crib speech and language play. New York: Springer Verlag.

Larsen-Freeman, D. (2012). On the roles of repetition in language teaching and learning. Applied Linguistics Review, 3(2), 195-210.

Lee, Y. S. (1983). Public management and case study methods. Teaching Political Science: Politics in Perspective, 11(1), 6-14.

Marley, S. (2007). EDPSY 505 exam #1 [Word document]. Retrieved from http://www.unm.edu/~marley/methppt/spring07/exam.doc

Moore, E. W., Erichsen, K., & Warta, S. (2014, May). Evaluating portfolios as a learning tool in a personality psychology course. Poster presented at the 26th Association for Psychological Science Annual Convention Teaching Institute, San Francisco, CA.

Murray, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art. International Journal of Artificial Intelligence in Education, 10, 98-129.

Newey, C. (1987). A case study approach to the teaching of materials. European Journal of Engineering Education, 12(1), 59-68.

Ohlsson, S. (1987). Some principles of intelligent tutoring. In Lawler & Yazdani (Eds.), Artificial intelligence and education, volume 1. Ablex: Norwood, NJ, pp. 203-238.

Rasinski, T. V. (1989). The case method approach in reading education. Reading Horizons, Fall, 5-14.

Renkl, A. (2002). Worked-out examples: Instructional explanations support learning by self-explanations. Learning and Instruction, 12(5), 529-556.

Royer, J. M., Tronsky, L. N., Chan, Y., Jackson, S. J., & Marchant, H. (1999). Math-fact retrieval as the cognitive mechanism underlying gender differences in math test performance. Contemporary Educational Psychology, 24(3), 181-266.

Rydland, V., & Aukrust, V. G. (2005). Lexical repetition in second language learners’ peer play interaction. Language Learning, 55(2), 229-274.

Saul, L. (2005). Evolution lab, natural selection simulation [Flash player simulation]. Available from http://biologyinmotion.com/evol/index.html

Schwartz, P. L., Fiddes, T. M., & Dempster, A. G. (1987). The case-based learning day: Introducing problem-based learning into a traditional medical curriculum. Medical Teacher, 9(3), 275-280.

Scully, A. L. (1984). The case method. History and Social Science Teacher, 19(3), 178-180.

Sottilare, R. A., Brawner, K. W., Goldberg, B. S., & Holden, H. K. (2012). The Generalized Intelligent Framework for Tutoring (GIFT). Concept paper released as part of GIFT software documentation. Orlando, FL: US Army Research Laboratory-Human Research & Engineering Directorate (ARL-HRED).

Taub, M., Azevedo, R., Bouchet, F., & Khosravifar, B. (2014). Can the use of cognitive and metacognitive self-regulated learning strategies be predicted by learners’ levels of prior knowledge in hypermedia-learning environment? Computers in Human Behavior, 39, 356-367.

Trevors, G., Duffy, M., & Azevedo, R. (2014). Note-taking within MetaTutor: Interactions between an intelligent tutoring system and prior knowledge on note-taking and learning. Educational Technology Research and Development, 62, 507-528.

UC Berkley, Berkley Center for Teaching & Learning. (2016). Course evaluations question bank. Retrieved from http://teaching.berkeley.edu/course-evaluations-question-bank

Vanlehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R., Taylor, L., ... & Wintersgill, M. (2005). The Andes physics tutoring system: Lessons learned. International Journal of Artificial Intelligence in Education, 15(3), 147-204.

Ward, W., Cole, R., Bolaños, D., Buchenroth-Martin, C., Svirsky, E., & Weston, T. (2013). My science tutor: A conversational multimedia virtual tutor. Journal of Educational Psychology, 105(4), 1115.

Weir, R. (1962). Language in the crib. The Hague: Mouton.

Wenger, E. (1987). Artificial intelligence and tutoring systems. Los Altos, CA: Morgan Kaufmann.

Wood, H., & Wood, D. (1999). Help seeking, learning and contingent tutoring. Computers & Education, 33(2), 153-169.

ABOUT THE AUTHOR

Samantha F. Warta, M.S. *is currently working as a research assistant in the Cognitive Sciences Lab at the University of Central Florida's Institute for Simulation and Training. Recently, she graduated summa cum laude with her Master's of Science in Modeling and Simulation from the College of Engineering and Computer Science at the University of Central Florida. Samantha received her B.A. in Psychology with a minor in Applied Statistics from Stetson University. She has previously published papers in the areas of college students' volunteering, serious games, human-robot interaction, and artificial social-cognitive architectures. Her current work includes the continued development and validation of a robotic perception scale, mental state attribution for effective social human-robot interactions, and the design of a novel robotic sensor system for the perception of social cues and signals.*

Intelligent Tutor System for Laboratory Testing for Febrile Rash Illness

Chaithanya Renduchitnala and Sarah Matthews
University of Central Florida Modeling and Simulation Program

INTRODUCTION

Outbreaks are localized events that can quickly overwhelm local health departments' ability to respond effectively. It is imperative that local departments can quickly prepare and deploy skilled responders once an initial case of a communicable disease is detected. Since outbreaks tend to occur sporadically, continuous training and preparation is necessary.

In the case of infectious diseases, proper adherence to protocols and any additional disease specific guidelines is critical to ensure the safety of the public. While licensed health professionals are trained on various aspects of infectious disease care, treatment and precautions; protocols tend to evolve and improve with time.

Medical errors due to insufficient or not current training by a medical professional may result in harm to the patient, consume additional resources, result in improper records or potentially contaminate clinical samples and even spread the infection. There are several psychological components such as prior knowledge of task, problems related to teamwork, communication, technology design, leadership and human decision making that determine the likelihood of error. Therefore, along with knowledge it is also important to understand attributes of the learner that may impact performance of task while being tested on the subject matter.

In this paper, we describe an Intelligent Tutoring System (ITS) utilizing the Generalized Intelligent Framework for Tutoring (GIFT) platform. The ITS was developed to enhance the local health authority's consultation and training responsibilities when responding to an infectious disease outbreak. The ITS assesses learner attributes via a pre and post knowledge assessment. The knowledge assessment checks on the learners' ability to recognize illness and order appropriate clinical testing to identify or rule out cases of communicable febrile rash illnesses in a timely manner. To order appropriate clinical testing, health care workers need to appropriately collect, store and ship diagnostic specimens to the state public health laboratory. Measles and Varicella are two communicable febrile rash illnesses that will be used to exemplify the processes in this paper.

Measles Sample Collection

Sample collection procedures during measles response outbreaks require significant understanding of the disease and protocols for collection and processing of clinical samples. The trainee must know the methods to detect measles infection and immunity. Measles virus can be detected from various samples by using cell culture techniques or molecular techniques. Measles identification methods are as follows; Serological assays including Immunoglobulin M (IgM) enzyme-linked immunosorbent assay (ELISA), Virus isolation and Reverse transcription polymerase chain reaction (RT-PCR). Throat (Oropharyngeal), nasal or NP (nasopharyngeal) swabs are the preferred samples for virus isolation or detection of measles RNA by RT–PCR. Synthetic swabs are recommended. Urine samples may also contain virus and when feasible to do so, collection of both samples can increase the likelihood of detecting the virus. Samples should be collected as soon after rash as possible or at the first contact with the suspected case. To assess for measles immunity in contacts (persons exposed to suspected case), the serological assays are utilized to test for IgM and IgG (Centers for Disease Control National Center for Immunization and Respiratory Diseases, 2015).

Varicella Sample Collection

Skin lesions are the preferred specimen for laboratory confirmation of Varicella virus. The swab is taken from the base of a wet lesion. Two filled in dime sized circles should be made on a plain glass slide and allowed to air dry. Two slides should be collected from each patient. Serum specimens are preferred to test for immunity (IgG). IgM testing may be performed on unimmunized subjects or on subjects with unknown immunity status. Blood specimens are collected using a vacutainer with a red stopper or serum separator tube.

Several methods including the isolation of varicella virus from a clinical specimen, direct fluorescent antibody (DFA), polymerase chain reaction (PCR) or detection of significant rise in serum varicella IgG by any standard assay meets the laboratory criteria for diagnosis. Specimens and the manner of collection for each may vary. Thus, the health professional needs to follow the exact procedure to safely and reliably collect and ship the clinical specimens. Additionally, demographic information about the subject and the clinical sample needs to be appropriately recorded on the specimen label (Centers for Disease Control National Center for Immunization and Respiratory Diseases (NCIRD), 2017). There are many areas in which errors can occur during the processing of clinical specimens. Training and refresher training can help reduce error.

Regulatory guidelines are updated by the State and Federal levels and can be located at the State Public Health Department and the Centers for Disease Control (CDC) websites. Training materials are generally available in various mediums from various medical sources including the State Health Department and CDC. Frequently, the responsibility for collating, curating and presenting the content to medical professionals falls on the local health officials. These duties may create additional burden especially during outbreak responses when the need for training is great and the resources to train may be allocated elsewhere.

Why use an ITS platform

Training or review provided at the start of outbreak takes time and resources away from the response. There are limited methods to assess the readiness of responders prior to their deployment to a response area. The department lead is often responsible for staff training and preparedness. However, in a classroom mode of training it is a challenge for even experienced teachers to personalize instruction and keep track of learner ability, prior knowledge and progress.

The Generalized Intelligent Framework for Tutoring (GIFT) platform allows for on demand personalized tutoring that can assess and tutor several health care workers without additional demands on local resources. The ability to author content, set up adaptive surveys based on learner performance and attributes may significantly improve learning outcomes for many health care workers ultimately improving the quality of response and delivery of care.

Learner attributes are intrinsic to the way each individual processes and assimilates information presented. The ability of a tutor to perceive learner attributes either by observation or by assessment and formulate content delivery based on these perceived attributes can greatly enhance learner engagement and improve learner outcomes. For this reason, we choose the Generalized Intelligent Framework for Tutoring (GIFT) as an Intelligent Tutoring System (ITS) for this tutorial. GIFT is a computer based tutoring system that incorporates a learner and a pedagogical module as part of the GIFT Authoring Tool

(GAT). These modules allow testing for adapting instruction to specific learner attributes by using specific learner states to select the next content to be presented to the learner (Sottilare, Grasser, Hu, & Brawner, June, 2015).

The ITS for Laboratory Testing for Febrile Rash Illness assumes the learner is a qualified healthcare professional that is familiar if not well versed in the content presented. The objective of the tutor is to provide refresher training on an as needed basis to the individuals who may not have collected the clinical samples recently. Therefore, the ability to assess prior knowledge and deliver content as needed prevents redundant training while identifying and tutoring only those individuals in need of additional support.

RESEARCH HYPOTHESIS

Human tutors have collated, curated, provided and when needed presented training material to ensure health care workers have appropriate level of background knowledge and the ability to apply that knowledge in response to a health emergency. However, this can be a time-consuming process, with little to no ability to verify if each individual learner has met all the learning objectives. The primary goal in developing the tutor is to ensure high quality materials are personalized and presented to a learner based on their prior knowledge, grit and ability to learn.

The research hypothesis that is proposed for this ITS will provide the learner with an adaptive training environment that requires little to no human tutor involvement in training. It will also provide an assessment of the learner's understanding and ability to apply appropriate laboratory criteria for the testing of selected febrile rash illnesses.

Surveys and Course Evaluation Questionnaire

Learner's knowledge improvement is assessed by evaluation of their pre-and post-performance surveys. The surveys are indicative of the ability of a leaner to learn the subject matter presented by the ITS. To complete the testing of the hypothesis, learners' perception of the ITS course is recorded by the course evaluation questionnaire.

RELATED RESEARCH

E-Learning

There are several aspects that need to be considered in the design of web-based intervention including, patient adherence, interaction, condition, feedback target behavioral outcome and its evaluation (Kelders, Kok & Ossebaard, 2012a). These concepts are encapsulated in the persuasive design framework deployed by Lehto and Onias-Kukkonen (2011) in a system to assist recovery from alcohol and smoking recovery. The persuasive system design paradigm when properly implemented in an appropriately designed system causes behavioral changes in trainees. Principles of persuasive system design have been shown to increase adherence with protocol in trainees. Many of the applications of persuasive system design aim to modify behavior in patients who suffer from chronic conditions such as diabetes. The persuasive system design framework considers reduction, tunneling and tailoring mechanisms. Reduction is breaking complex behavior into its simplest form. Tunneling is putting the user through set experiences or processes designed to persuade the user. Tailoring is personalization of the content to the user's personality and needs. Additionally, the framework evaluates the system's ability to empower patient self-monitoring by the simulation of scenarios to rehearse the impacts of certain behavior. Single user interactive prompts such as praise, rewards, reminders, suggestion and liking are considered during system design for their impact on behavioral change. Social support aspects such as learning along with peers, either by comparison, cooperation or completion are considered along with other parameters such as recognition of achievement by peers. These web-based personalized medical simulation and serious gaming system have made the possibility of low cost patient specific care and protocol compliance system that promise to improve clinical quality outcomes and measures in the future a reality (Kelders et al. 2012b).

Serious Gaming

Studies have shown that serious gaming can enhance learning outcomes and improve real world performance when used to train emergency medical responders (Knight et al., 2010). Several serious games that immerse the trainee in various disaster or surge event scenarios have been found to positively impact trainee preparedness for the real event. These scenarios include mass causality events triage (Pelaccia et al., 2009 ; Knight et al., 2010) mass causality burn events (Kurenov et al., 2009) and disaster response drills that utilize virtual reality technologies (Breslin et al., 2007). Recently, serious gaming tools especially online multi-player games have proven to be effective in planning for response (Breslin et al., 2007) while task simulators for the purpose of training are now well established in health care (Craft, Feldon & Brown, 2014).

Merrill's Component Display Theory

Component display theory (CDT) due to its precise matching of content classification with leaner performance is well suited for computer based training. The theory postulates that instructional outcomes can be classified on two dimensions: student performance and subject matter content. The CDT ties together performance categories with content categories and tests the learners' ability to understand and apply principles broadly when required. Performance categories include the ability to recall (remember), apply (use) and ultimately identify new situations not described in the tutor and apply concepts explained in the tutor (find). Content categories include the ability to retain facts, develop conceptual understanding, and describe task procedures (sequence). These interrelationships are presented in a performance / content matrix that can be developed for any given cognitive learning scenario. A limitation of the CDT is that it does not assess psychomotor tasks and affective objectives (Merrill, 1983).

Instructional presentation in the ITS are comprised of a series of discrete survey questions, displays and media presentations. The presentation of the material has two dimensions: content mode (generality or instance) and presentation mode (expository or inquisitory). The tutor developed is based on the primary presentation forms that combine generality with Expository (rules) and then with Inquisitory (recall) and test the generalities using instances with Expository (examples) and Inquisitory (practice). The survey object available in GIFT is an implementation of this theory (Sottilare, Grasser, Hu, & Brawner, 2015).

ITS DESIGN AND STRUCTURE

Tutor Process Overview

We developed an ITS using GIFT to facilitate training of medical staff in the laboratory testing of febrile rash like illnesses. At the start of the course the learner is shown an introduction which advises the learner of the concepts that are covered in the course. After the introduction, the learner will be asked to complete an 11-question survey. The survey is a self-evaluation to ascertain the learner attributes as it relates to Knowledge, Prior knowledge, Grit, Skill and Learner Ability. The questions are in the format of multiple choice (2), True/False (1), and Likert 4 point scales. The multiple-choice questions are gathering information on the learner demographics and the two questions posed are “Please choose which age range that best describes you” and “Please use the range that best describes how many years you worked in a healthcare setting”.

The learner completes the learner attribute survey and then is asked to complete a 25-question knowledge assessment. This survey is structured around the four concepts the ITS covers. The pre-test will be used in comparison with the post knowledge assessment to ascertain whether learning occurred. It will also be used to adapt the tutor so that the appropriate content is presented to the learner based on the performance of the learner. Regardless of the performance on the pre-test, all learners will be presented with the measles PowerPoint and the varicella PowerPoint. These slide sets contain the information on the four concepts and are set in the Rule Phase of the Adaptive Courseflow object. A schematic of the course flow is shown in Figure 1.

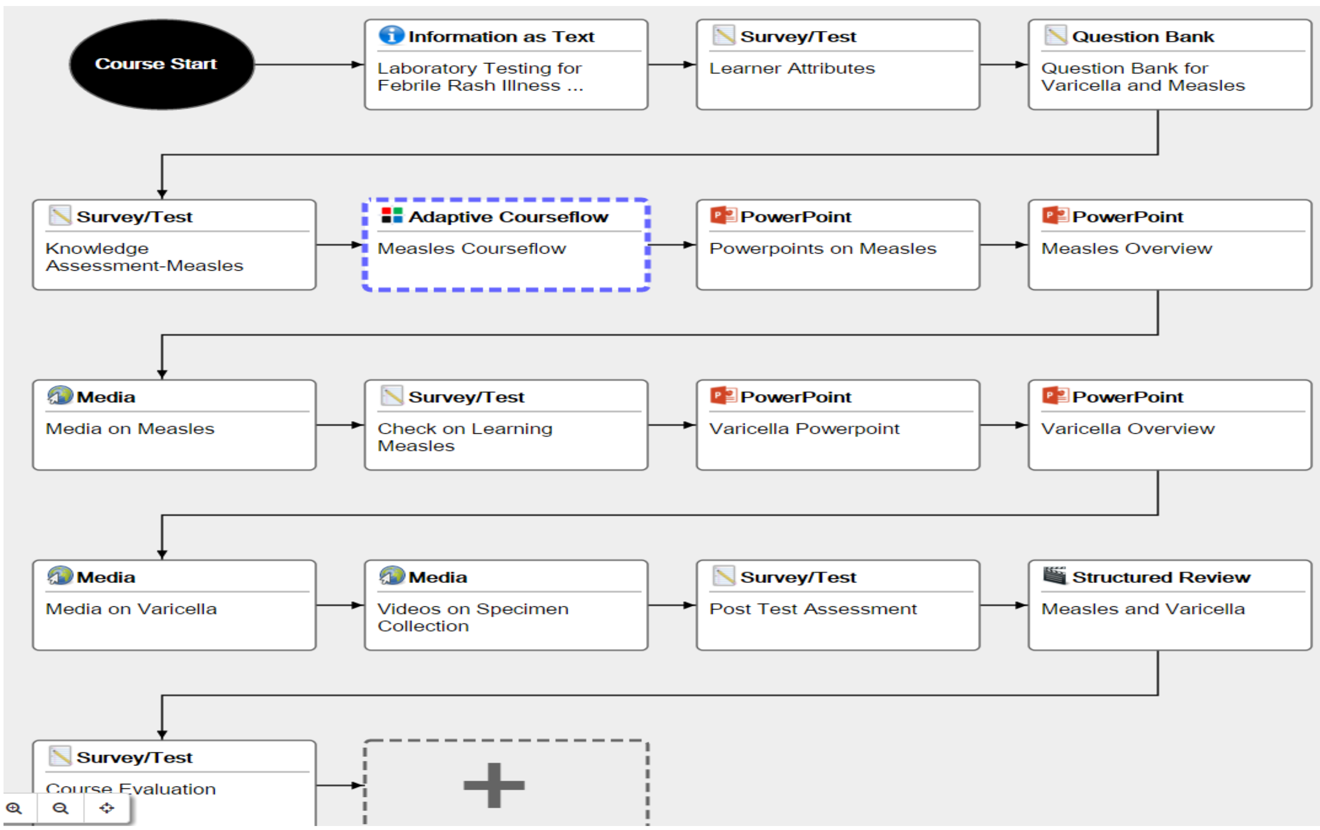


Figure 1: Laboratory Testing for Febrile Rash Illness Course Flow

If the learner is scored as a Novice or Journeyman from the Pre-Test assessment, he will be presented with content from the Example Phase which includes Overview PowerPoints on Measles and Varicella and 3 pieces of Media on each topic. If the learner scores as an Expert, after viewing the Measles PowerPoint, he will go immediately to the Check on Learning for Measles and then go into the Varicella PowerPoint. If the Check on Learning criteria is not met, the learner will be presented with the Rule Content again and have the option to select the Media Content to review. This will occur until the learner can successfully demonstrate understanding on the assessment surveys. Once the learner has completed reviewing the tutor content, he will be asked to complete a 25-question post knowledge assessment. This assessment is a duplicate of the pre-assessment.

Finally, the learner is presented with the Course Evaluation Survey. This is a 13-question survey that is composed of 5 point Likert Scale questions that range from Strongly Agree to Strongly Disagree and Free text questions. The purpose of this survey is to receive feedback from the learner on the ease of use of the system, the tutor content and if learning the content will led to a change in behavior.

Content Design

The guiding principle in our design of the current version of the ITS will be to test our proposed research hypothesis. Specifically, we will seek to ascertain if an ITS system can substitute a human tutor or minimize the human tutors' involvement for a public health response. As a first step the key concepts were identified as: Lab testing for Measles, Specimen and Lab Collection for Measles, Lab testing for Varicella (Chickenpox) and Specimen and Lab Collection for Varicella (Chickenpox). These concepts lend themselves to testing and assessment based on the principles in component display theory. We collated content and developed presentation paradigms for the expository rule and example phase. A decision was made to limit the current iteration to these four concepts and test the system before incorporating any additional concepts. Each of these concepts has content and media files associated with it that were collated from material that is generally made available to health care professionals.

The system was authored to provide all the necessary background information when needed based on the leaners' performance.

Survey Test Design

Two intake surveys were designed to assess the learner's attributes and prior knowledge. The adaptive course flow combines the rule and the example phase with the inquisitory recall and practice phase. For this tutor, we identified questions for the recall phase but did not include the practice phase as we were primarily interested in testing the ability of the learner to demonstrate retention and ability to apply key concepts that are covered in the expository phases.

DISCUSSION AND RECOMMENDATIONS

ITS design experience

The initial phase of the content development was focused on the development of material in appropriate format and medium to illustrate the key concepts. Once the expository content was selected and developed, as needed surveys were designed to test the proposed research hypothesis. The initial leaner attribute and prior knowledge survey were used to classify the user as a novice, journeyman or expert. This classification was used to drive the learners experience in the ITS environment. However, the system was developed with minimal ability to test the overall course flow. This limited our ability to iteratively improve the course flow and take full advantage of the adaptive course flow module.

A question bank comprising of the questions that test the knowledge of the learners on the concepts was developed. Individual questions were rated as easy, medium and hard and associated with specific concepts. Progression of content presented to the learner is determined by the performance of the learner on the questions. The number of easy, medium and hard questions that require correct responses is set in the recall phase of the adaptive course.

The post assessment evaluated the short-term retention of material covered by the tutor. Finally, the course evaluation survey captures user perception of the content, questions and the tutor system. The post assessment and the evaluation survey are designed to validate the research hypothesis.

Recommendations

An iterative design process separating content development and instructional design will help to fully leverage the abilities of the GIFT tutor and ensuring the tutor is consistent with the CDT. A formalized tutor development process will help the authors to develop testable tutors that are grounded in theory.

Content Development Process:

The content development process can be time consuming if not planned at the outset. This section outlines three recommendations that should help in development of course content.

- Revision of the material and the survey to reduce total time it will take to complete the mandatory training. Once an initial draft of the material is prepared it is imperative to compute the total time it will take for the material to be presented to the learner in the ITS. If the total time exceeds thirty minutes a reduction of the material is recommended. In our experience including essential material for each concept for the initial draft of the course is helpful.
- Division of the content into more discrete sections that correspond to specific learner objectives. It is helpful to set a finite maximum time for review of each section as per recommendations of the CDT. Base scoring on time taken to complete the assessment survey, especially the post assessment survey.
- Classification of content and presentation element into primary presentation forms. The GIFT system can be improved by creating authoring templates or wizards for content creation. The wizard can, for example, guide the author through the content development process steps and provide recommendations for what needs to go into the rule, example, recall and practice phase.

Instructional Design Process:

The following five recommendations address the instructional design process for using GIFT for instruction.

- Develop a performance – content matrix before implementation in the adaptive survey object. Select objectives based on intended performance- content level. The adaptive survey object requires planning before adding questions and associating questions with content. It will help to develop on paper a performance objective in terms of number of easy, medium and hard questions a learner needs to answer correctly. In this outline, include learner objectives associated with each performance measure to be tested in the adaptive survey. This will save time, prevent errors and allow more author control during the creation of the adaptive survey object. This will ensure that the adaptive survey object tests the learner on all the objectives and concepts that are covered in the content.
- The GIFT system can be improved by providing a course dashboard that provides an overview of the documentation of prompts, listing the number of items, evaluation of survey questions for divergence and difficulty if based on the same concepts.
- Consider timing and delay of post assessment or development of staged post assessment to assess impact on long term retention.
- Development of novice, journey man, expert criterion.

- Resolve other violations or infractions from the CDT.

System Development and Game based module

- Development of a game based practice module (the practice will test instances of recall, procedure and location (this is a test that extends the CDT theory and would be an affective test / familiarization)

It is our intent to conduct a study to compare the GIFT tutor with a human tutor in terms of effort level required to train a group of learners for the course developed. There were technical issues at the time of writing of this paper which hindered full execution of the study. However, approvals for the study and recruitment of subjects has been initiated. Finally, a formal course design process will support creation of content by multiple authors. In many cases, courses are developed by author groups and a formal design process may help to accelerate the adoption of intelligent tutors in the author community.

There has been a dramatic increase in the number of e-learning and adaptive learning platforms tutors developed with consistent design based on theory that will allow for comparison between tutors. A future study concept would be to provide the same material using another freely online MOOCS tutor such as udemy.com and compare learning outcomes of that learner group to the group using GIFT.

REFERENCES

Breslin, P., C. McGowan, B. Pecheux & R. Sudol (2007). Serious gaming. Advanced computer simulation games help to transform healthcare and disaster preparedness. *Health Manag Technol,* 28**,** 14, 16-7.

Centers for Disease Control National Center for Immunization and Respiratory Diseases (NCIRD), D. o. (2017, January 9). *Chickenpox (Varicella) Collecting Specimens for Varicella Zoster Virus (VZV) Testing*. Retrieved from Centers for Disease Control: https://www.cdc.gov/chickenpox/lab-testing/collecting-specimens.html

Centers for Disease Control National Center for Immunization and Respiratory Diseases, D. o. (2015, May 4). *Measles (Rubeloa) Specimens for Detection of Measles RNA by RT–PCR or Virus Isolation*. Retrieved from Centers for Disease Control: https://www.cdc.gov/measles/lab-tools/rt-pcr.html

Craft, C., D. F. Feldon & E. A. Brown (2014). Instructional design affects the efficacy of simulation-based training in central venous catheterization. *Am J Surg,* 207, 782-9.

Kelders, S. M., R. N. Kok & H. C. Ossebaard (2012a). Persuasive System Design Does Matter: A Systematic Review of Adherence to Web-Based Interventions. *Journal of Medical Intenet Research,* 34.

Kelders, S. M., R. N. Kok, H. C. Ossebaard & J. E. Van Gemert-Pijnen (2012b). Persuasive system design does matter: a systematic review of adherence to web-based interventions. *J Med Internet Res,* 14**,** e152.

Knight, J. F., S. Carley, B. Tregunna, S. Jarvis, R. Smithies, S. de Freitas, I. Dunwell & K. Mackway-Jones (2010). Serious gaming technology in major incident triage training: a pragmatic controlled trial. *Resuscitation,* 81, 1175-9.

Kureov, S. N., W. W. Cance, B. Noel & D. W. Mozingo (2009). Game-based mass casualty burn training. *Stud Health Technol Inform,* 142**,** 142-4.

Lehto, T. & H. Onias-Kukkonen (2011). Persuasive features in web based alcohol and smoking interventions: a systematic review of medical literature. *Journal of Medical Internet Research,* 13, 3. doi:10.2196/jmir.1559

Merrill, D. M. (1983). The Descriptive Component Display Theory. In C. M. Reigeluth, *Instructional Design Theories and Models: An Overview of their Current States* (p. 360). Routledge; 1 edition (August 1, 1987).

Pelaccia, T., H. Delplancq, E. Triby, J. C. Bartier, C. Leman, H. Hadef, N. Meyer & J. P. Dupeyron (2009). Can teaching methods based on pattern recognition skill development optimise triage in mass-casualty incidents? *Emerg Med J*, 26, 899-902.

Sottilare, R. A., Grasser, A., Hu, X., & Brawner, K. (June , 2015). *Design Recommendations for Intelligent Tutoring Systems.* Orlando : U.S. Army Research Laboratory

ABOUT THE AUTHORS

Chaithanya Renduchitnala *is currently a graduate student in the University of Central Florida's Modeling and Simulation Program with a research focus on M&S training tools and methods for Social Learning Networks.*

Sarah Matthews *is currently a graduate student in the University of Central Florida's Modeling and Simulation Program with a research focus on M&S training tools and methods for the applied Public Health Epidemiologist.*

THEME VII: ANALYTICS AND EFFECTIVENESS MEASURES

Modeling Training Efficiency and Return on Investment for Adaptive Training

Gregory A. Goodwin[1], Jong W. Kim[1], James Niehaus[3]
ARL HRED1, Charles River Analytics2

ABSTRACT

Adaptive training promises more effective training by tailoring content to each individual. Where non-adaptive training may be just right for one segment of the student population, there will be some students that find it too easy while others find it too difficult. Another, often ignored benefit of adaptive training, is improved training efficiency by minimizing the presentation of unnecessary material to learners. One implication of this is that intelligent, adaptive training should require less time to train a population of learners to a given level of proficiency than non-adaptive training. The gains in efficiency should be a function of several factors including learner characteristics (e.g., aptitude, reading ability, prior knowledge), learning methods employed by the adaptive training system, course content (e.g., difficulty and length, adaptability), and test characteristics (e.g., difficulty, number of items). This paper describes the development of a predictive model for training efficiency based on those factors and how it could be integrated into the Generalized Intelligent Framework for Tutoring (GIFT) architecture. How this model supports return on investment decisions for authors is also discussed.

INTRODUCTION

The Generalized Intelligent Framework for Tutoring (GIFT) is an open-source, modular architecture developed to reduce the cost and skill required for authoring adaptive training and educational systems, to automate instructional delivery and management, and to develop and standardize tools for the evaluation of adaptive training and educational technologies (Sottilare, Brawner, Goldberg, & Holden, 2012a; Sottilare, Goldberg, Brawner, & Holden, 2012b). By separating the components of ITSs, GIFT seeks to reduce development costs by facilitating component reuse.

Meta-analyses and reviews support the claim that intelligent tutoring systems (ITS's) improve learning over typical classroom teaching, reading texts, and/or other traditional learning methods. (Dynarsky et al. 2007; Dodds and Fletcher 2004; Fletcher 2003; Graesser et al. 2012; Steenbergen-Hu and Cooper 2013, 2014; VanLehn 2011). In fact, ITSs have been shown to improve learning to levels comparable to Human tutors (VanLehn et al. 2007; VanLehn 2011; Olney et al. 2012).

While improved training effectiveness is certainly a benefit of ITS technology, another important benefit is improved training efficiency over one-size-fits-all training. The goal of an ITS is to identify the gaps in knowledge specific to each learner so that training can focus on filling just those gaps. One of the problems of one-size-fits-all training is that to insure all trainees can comprehend the instruction, it must be developed for trainees with the least experience, knowledge, and aptitude. Though less costly to develop, the material is presented a pace that is slow and that includes content not needed for more experienced, higher aptitude trainees. An ITS would be expected to reduce the time needed to deliver training to such trainees.

The reduction in time to train (i.e., improved acquisition rate) is an important metric because reductions in training time represent cost savings. This is especially true for military trainees who are paid a salary.

Reductions in the time needed to train those trainees save salary costs for both trainees and instructors. For large-volume courses, those savings can be substantial.

All of this highlights the need for a means to model and predict training efficiency gains (i.e., time saved) by ITSs generally and GIFT specifically. Having the ability to model time saved by the use of adaptive, intelligent training, as compared to existing or non-adaptive training would have benefits throughout the lifecycle of a course. During the design of new training, the training developer could more easily make decisions about the relative costs and benefits of adding adaptive features. For example, adding extensive remedial training for easy-to-understand concepts may benefit such a small percent of the population of learners, that the net reduction in training time would be too small to make those features worth the cost of development.

During training delivery, actual trainee data could be used to verify and/or improve the model. For example, suppose the model assumed that learners with an aptitude above criteria A would have a 95% probability of understanding concept B without needing any remediation. Learner data could then be used to validate or adjust that probability. This improved model could then be used to better determine the true time-savings of the course when delivered by GIFT.

During training evaluation and refinement, the disparity between predicted and observed training outcomes could be used to refine the training. For example, if a segment of training proves to be more difficult than anticipated for a group of learners, it is possible that the training segment should be refined or redeveloped.

An example of such a model was developed by McDonnell Douglas (1977). This model incorporated predictor variables in four broad categories: course content (e.g., difficulty, length of content), instructional design (e.g., instructional strategies/techniques), test characteristics (e.g., difficulty, number of items), and trainee characteristics (e.g., aptitude, motivation). The model predicted about 39% of the variability in trainee's first-attempt lesson time for self-paced computer-based instruction.

To understand how GIFT might begin to model and predict training time for learners, it is necessary to understand how training is adapted by this system. GIFT is a framework that modularizes the common components of intelligent tutoring systems. These components include a learner module, an instructional or tutor module, a domain module, and a user interface. One of the main motivations for creating this framework was to lower the cost and labor needed to create intelligent tutoring systems by facilitating re-use of components and by simplifying the authoring process (Sottilare et al., 2012a).

GIFT adapts training using the learning effects model. At the first point of this model, learner data informs the learner state in the learner module. The learner module receives assessments from both sensors and the domain module. The learner state is used to determine the appropriate instructional strategy by the tutor module. The instructional strategy is then interpreted by the domain module and used to determine the domain specific learning activities needed to instruct the learner in that domain. The responses of the learner to that activity then update the learner module which starts the cycle over again.

Developing a predictive model in GIFT is not a straightforward process given the ways that training is adapted to each individual. We should note that our goal is not to predict the single path that a trainee would be expected to take through a specific course, but rather the probability associated with all possible paths through the training for a given learner. From that we can determine the range and distribution of times that would be expected for that learner to complete the training. Taking this one step further, we could apply this to a population of learners and predict the range and distribution of the time for that population to complete that training.

The development and integration of a probabilistic model for predicting time to train into the GIFT architecture is currently in the first phase of a three phase plan. In this paper, we describe work being done in the first phase. In this phase we are developing the structure of the Bayesian probabilistic model, identifying factors that are expected to impact training time, and mapping those to a specific course delivered by GIFT. In the second phase, we will integrate this model into the GIFT framework and develop the user interface to allow for authoring of new predictive models for other GIFT courses. In the third phase of the work, we will empirically validate the predictive model in GIFT and make adjustments to try to improve it.

METHODS

This section describes our method for modeling adaptive training content and predicting distributions of completion times for both individuals and groups using the GIFT excavator trainer as an example. This course is available with public version of GIFT. The training content includes text, images, video demonstrations, and practice opportunities in a virtual simulator making it a good example of the kind of adaptive training that GIFT can deliver.

An Adaptive Training Course in GIFT: Excavator Training

The excavator training course (Army Research Laboratory, 2015) consists of MS PowerPoint slides with instructional information and questions, and a 3D simulation environment for practice. The excavator training starts with a welcoming message and a set of survey questions that obtain the learner characteristics of motivation, grit, and self-regulatory ability. The GIFT tutor, then, presents the concepts of rules to control the excavator (i.e., Excavator, Boom, Bucket, Arm, and Swing), and corresponding examples. Figure 1 shows the overall structure of the excavator training contents.

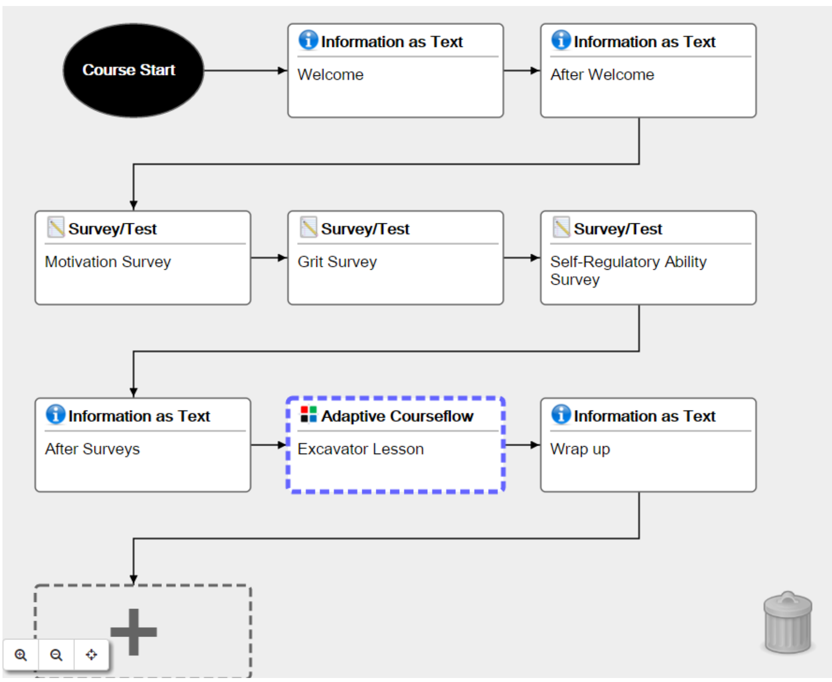


Figure 1. The overall structure of the excavator training course and the adaptive courseflow of the Recall phase in GIFT.

The Adaptive Course Flow object in GIFT (formerly known as the Engine for Management of Adaptive Pedagogy – EMAP, e.g., Sottilare, 2014; Goldberg, 2015) supports adaptive capabilities for training based on the Component Display Theory (CDT, Merrill, 1983). The CDT supports a general framework of skill training that progresses through two types of learning activities, each with two categories: expository (rules and examples) and inquisitory (recall and practice). According to Merrill, learners should progress through these four quadrants in order starting with rules (presentation of general principles), then to examples (presentation of a specific instance), then to recall (declarative knowledge test of the trainee's comprehension), and finally to practice (opportunity for the trainee to perform the skill). By sorting learning activities into these four quadrants, adaptive training systems like GIFT can apply the CDT to any domain as long as content for that domain is so labeled.

Modeling the Content of Adaptive Training

To model the content of adaptive training, we use the *Methodology for Annotated Skill Trees* (MAST) skill trees. The "skeleton" of the skill tree breaks down entire procedures into constituent steps, tasks, and subtasks. Annotations are added to the procedure model. For example, consider completing a set of questions in the excavator tutor that features hints and feedback. This step includes tasks for reading the introduction to the problems, each problem, reading hints, and reviewing feedback. Critical for adaptive training, the MAST procedure model represents not only the base procedure of answering each question correctly without hints, but also the optional hints and feedback steps, variations, and multiple potential paths among questions as chosen by GIFT. Annotations within the MAST skill tree include the following additional information for each step, task, and subtask.

- *Information Elements:* Information or knowledge needed by the trainee to perform the actions required by the skill tree node. These requirements are commonly called the "knowledge map" in ITS literature. In the example of completing a set of GIFT questions, this is the knowledge used to answer the question correctly.

- *Instructional Resources:* Resources to teach the skills needed to perform the actions required by the node. In the question example, these are pointers to additional training content.
- *Skill Priorities*: Ratings of the difficulty and criticality of the skills needed to perform the actions required by the node. These ratings enable training systems to prioritize skills for training and optimize ROI. In the question example, ratings express the criticality of answering the questions correctly to the overall learning goals.

- *Assessments*: Methods of assessing the skills required by the node. These methods enable training systems to determine trainee ability. In the question example, assessment methods include secondary measures of trainee cognitive workload, motivation, or affect that may influence completion time.

- *Decision Making Models*: Computational models of how the procedure steps, tasks, and subtasks are chosen and ordered. These models enable some of the adaptation logic to be represented in the skill tree. In the question example, these models encode the rules for providing hints, providing feedback, and selecting the next question.

- *Completion Time Data:* describes a distribution of completion time based on past data or an estimate of completion time based on type. This data will be used to train the prediction algorithms

We use a probabilistic model to represent the different factors and instructional strategies that impact the completion time of a MAST module, as well as probabilistic inference techniques to determine a distribution of a course completion time. Not only must our model represent relationships between variables and paths in the MAST skill tree, but it must also recognize and model the impact of time as well; many variables can change as the trainee is completing a training module. Building this model consists of two basic steps: developing a model that estimates the completion time for nodes in the MAST skill tree, and temporally linking these models together to enable inference of the entire module completion time.

Figure 2 shows part of an example model for estimating the completion time of a node in a MAST skill tree. This example shows some contributing factors that could be used by PAST Time to estimate the time it takes for a trainee to read the text on the slide. There are also variables that estimate the time to process the pictures and audio on the slide, but that these have been omitted from this example for brevity.

The model includes a Reading Time variable, which represents the time it takes for the user to read the text. The value of this variable is a function of the amount of text on the slide, the speed at which the trainee can read the text (Read Speed), and the current alertness of the trainee (Fatigued). These relationships are probabilistic. For example, if a trainee normally reads at 100 words per minute, there are 100 words in the text, and the trainee is tired, the reading time of the trainee could be distribution uniformly from 1 to 2 minutes. The reading speed of the trainee is also a non-deterministic variable that depends on how much prior knowledge the trainee possesses about statistics about how fast the general population of trainees read.

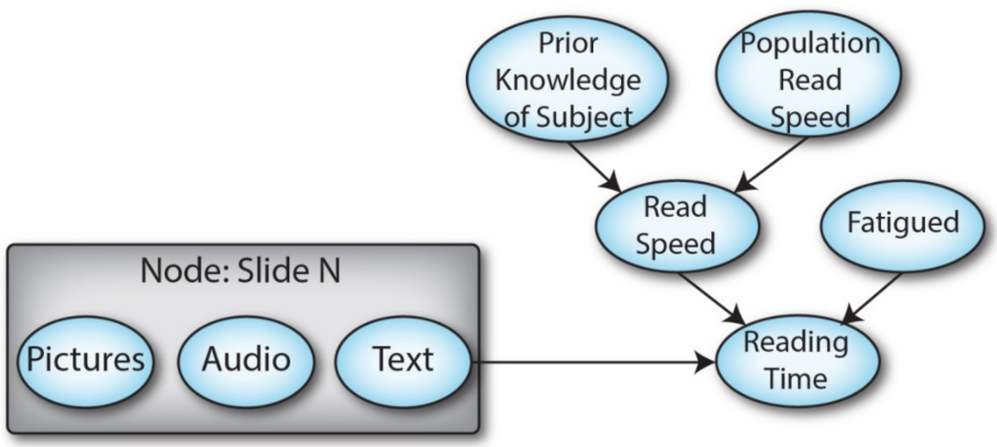


Figure 2: Example model for estimating the time to read Text on a Slide node.

One of the benefits of building a probabilistic model to represent the completion time is that not all of the information in the model is needed to estimate the completion time. For example, if we know how much prior knowledge the user has about the subject (for example, from a pre-instruction questionnaire), we can post that knowledge as *evidence* to the model that would be taken into account when estimating the completion time. If we do not possess that information, we can treat the variable as *latent* and use a prior distribution to represent the state of the variable. For example, we can estimate that only 20% of trainees taking the course have prior knowledge of the subject. These prior distributions can be estimated from the literature review or expert knowledge, and then *learned* over time based on the outcomes of actual testing.

Figure 2 shows a portion of a MAST skill tree for the excavator training GIFT course. This skill tree focuses on the information elements that most heavily influence the completion time. On the left, the overall course on Excavator is the root of the tree structure. Its children are the different topics covered by the course, including the Boom Movement topic. This topic features a number of slides with Pictures, Audio, and Text components. Individual trainees may vary in the amount of time they spend examining the Pictures, whether or not they listen completely to the Audio, and the amount of time taken to read the Text. Trainees may also choose to view optional Slides explaining concepts that they may not be familiar with, adding more time. If trainees fail to demonstrate sufficient knowledge in the quiz or fail to complete the simulation tasks appropriately, they are sent back to the beginning of the Boom Movement topic on Slide 1, adding significant time to completion of the course. This model may be expanded to represent a maximum number of failures before the trainee either moves to a different topic or ends the course.

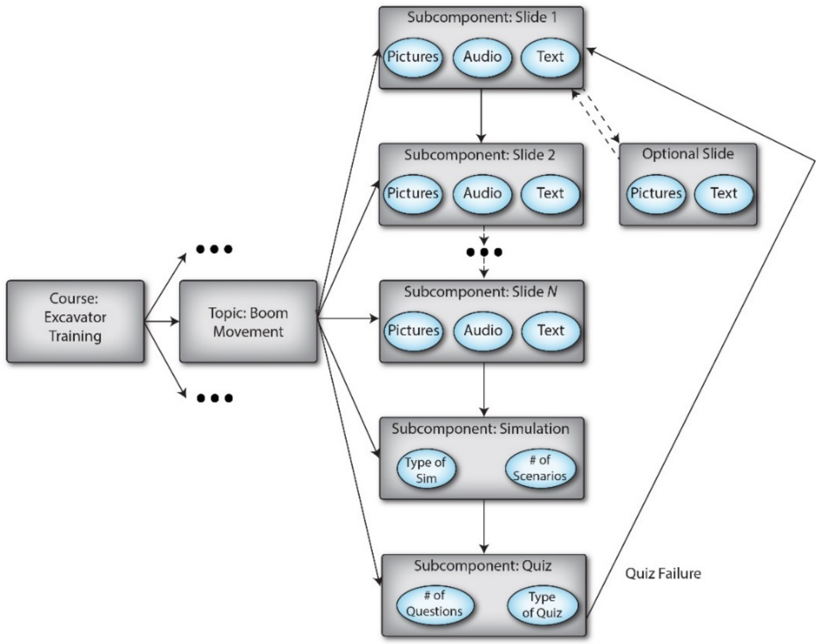


Figure 3: High-level design of a MAST skill tree of a GIFT module with representations of individual instructional elements, branching content, and variables that influence completion times.

After reviewing the Slides, the trainees are asked to practice their skills in Simulation. The MAST model of the simulation can be either a complex procedure describing the steps needed to complete the scenario and optional steps that may or may not contribute to the overall goal. The MAST simulation model may also be simple, representing just the type of simulation and the number of scenarios. To save modeling time and effort, these MAST models are constructed with only the level of detail needed to sufficiently and accurately predict the completion time.

Once these probabilistic models are defined, they can be used to compute a distribution over the course completion time. To generate this distribution, a modeler first provides knowledge about a trainee, group of trainees, or a module as evidence to the model. This could be statistical information obtained from the trainees from a pre-course questionnaire, or data obtained from prior training. Then, given the posted evidence, the user can apply standard probabilistic inference techniques (e.g., variable elimination, importance sampling, Metropolis-Hastings, support computation, most probable explanation (MPE), and particle filtering) to generate a distribution over the completion time of the module. These specific methods are included in the Figaro libraries. Statistical moments of this distribution (e.g., mean and variance) can be easily computed and presented to a module designer.

A significant advantage of combining this probabilistic modeling with the MAST skill tree representation is the capability to ascribe time to individual models, and perform "what if" analysis by adding or removing components. For example, a node for a module requiring detailed arithmetic may take little time in and of itself, but it may be fatiguing, causing significant downstream effects in terms of overall training completion time.

RESULTS

Implementing the Adaptive Training Models

The probabilistic model is being implemented using Charles River Analytics' open source probabilistic programming language, Figaro™ (Pfeffer 2012), to construct and learn probabilistic models of the relationships between these factors. The use of Figaro will greatly simplify the authoring of these models which can be complex and require a high degree of experience by users who may not be experts in probabilistic reasoning.

Figure 3 shows an example Figaro program that creates the completion time model for the node slide shown previously in Figure 2. Note that the probabilities and values in this program are notional. First, we define the amount of text in the node as 1000 characters. Then, we define two latent variables, one representing the prior knowledge of the trainee and the other representing typical reading speeds. In this case, we specify that a trainee has prior knowledge with 0.2 probability, and the trainee's reading speed is normally distributed around 100 characters a second. Next, we define the actual reading speed of this trainee. In this example, if the trainee has prior knowledge of this subject, we increase their reading speed by a value normally distributed around 50 characters a second. We next represent the fatigued state of the trainee (0.4 probability that the trainee is fatigued). Finally, we define the reading time of this node as the amount of text divided by the reading speed of the trainee; if the trainee is fatigued, however, we assume they can only read at 50% capacity. To use this model to estimate the completion time of the module, we use Figaro's built-in importance sampling algorithm to sample the model and print the distribution over the reading time variable. Observe that invoking an inference algorithm to estimate the completion time is a single line of code, and any other Figaro inference algorithm can be substituted into this program with no other changes.

```
val text = Constant(1000.0)

  val priorKnowledge = Flip(0.2)

  val populationReadSpeed = Normal(100.0, 50.0)

  val readSpeed = If(priorKnowledge,

    populationReadSpeed ++ Normal(50.0, 25.0), populationReadSpeed)

  val fatigued = Flip(0.4)

  val readingTime = If(fatigued,

    text / (readSpeed * Constant(0.5)), text / readSpeed)

  val algorithm = Importance(10000, readingTime)

  algorithm.start

  println(algorithm.distribution(readingTime))
```

Figure 4: Figaro program that models reading time of a Slide node.

Figaro probabilistic programming is useful in this context for a number of reasons: We can automatically build a model given a specification of the MAST skill tree, the trainee model, and a set of known relationships. Prediction based on the model is already coded in Figaro's inference algorithm, so additional effort is not required to use the model. Figaro supports the creation of dynamic Bayesian networks that model the temporal processes of variables, simulating fatigue and practice effects. We can continuously learn using these models; the probabilistic programs are flexible enough to update relationships between variables based on historical or dynamic data. Figaro's encapsulation mechanism enables easy creation of reusable components. Trainee models and MAST skill trees can be reused for future prediction models. It is embedded in a general purpose language, Scala, which allows the creation of front end graphical interfaces that can edit and invoke the models created in Figaro.

Figure 5 shows the results of running this Figaro model. The distribution of reading times has three modes. At about 7 seconds, individuals that have prior knowledge and are not fatigued read the slide quickly. At 10-11 seconds are individual that have no prior knowledge and are not fatigued. At 20-21 seconds are individuals without prior knowledge and who are fatigued, reading slowly to absorb more information. An instructor may use a model like this one to examine how individual slide contents may be processed by a class of students, and make small changes to the presentation to increase learning efficiency.

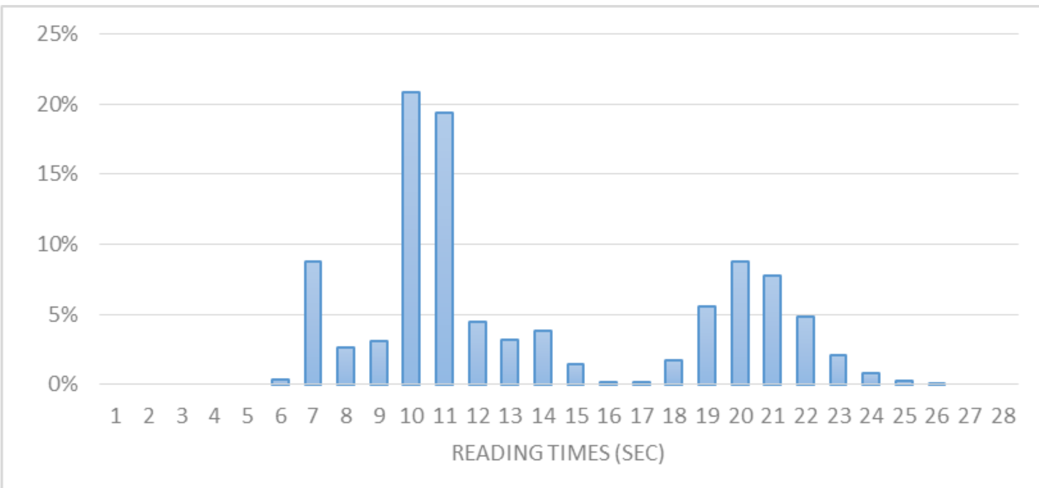


Figure 5: Probability Density of Reading Times for One Slide.

Figure 6 shows the probability density of reading times over three slides with the student having increased chance of fatigue (40%, 45%, and 50%) on each successive slide. In this simulation, only a small portion of the students are in the fastest group, completing three slides in about 20 seconds. The bulk of the students range from 25-55 seconds for these three slides, with three modes in this range covering the combinatorics of prior knowledge and different possible fatigue states on each slide. Also, a significant portion of the students takes longer than 55 seconds, with a possibility of up to 76 seconds to complete. An instructor can use this model to examine the differential effects of fatigue, prior knowledge, and reading speeds of a heterogeneous group of students, and adjust the learning content or course expectations accordingly.

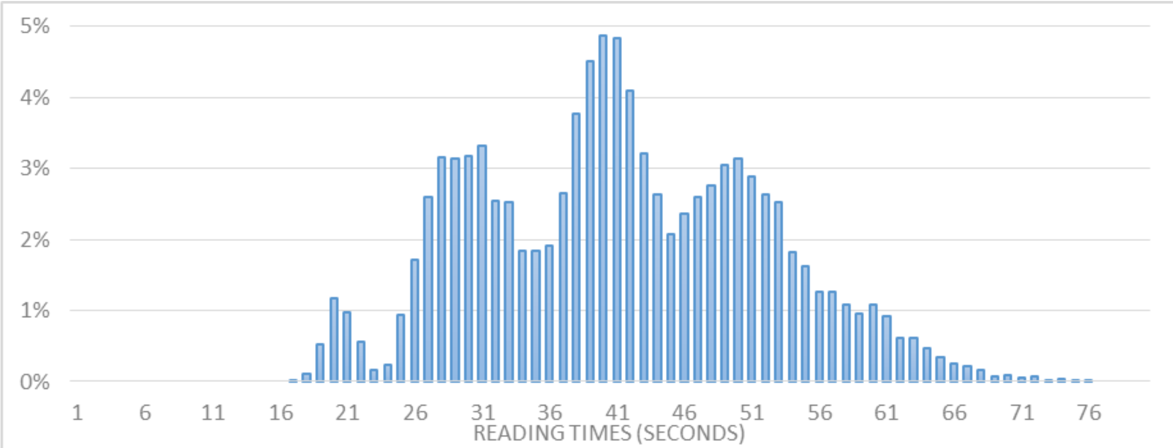


Figure 6: Probability Density of Reading Times for Three Slides with Increasing Chance of Fatigue.

This modeling can reveal underlying properties of the adaptive learning content that may be counter-intuitive at first glance. For example, the most likely reading speed of a single slide (according to the first model) is about 10 seconds. For three slides, one might assume 10 * 3 = 30 seconds, but the distribution in Figure 6 shows the mean of the predicted time about 41 seconds with significant standard deviation. Allotting only 10 seconds on average per slide in a course would prevent about two-thirds of students from completing all of the course content.

The adaptive training content with significant remedial steps has a much wider variance of completion times. We hypothesize that retraces through previous material (e.g., reviewing the boom operation slides) will be performed much faster than the initial trace. Trainees may also be able to optimize their reading and comprehension strategies if they know how they will be tested and what the consequences for failing are. Therefore, later sections in an adaptive training course (e.g., excavator bucket handling after boom handling) may have significantly different variable interactions than earlier sections, as trainees learn the training structure.

DISCUSSION

We believe that including a capability to predict training time for trainees in GIFT has several significant advantages for accelerated learning. First, it facilitates return on investment (ROI) calculations by enabling the author to determine training time reductions resulting from the addition of adaptive features. Second, it provides a means for GIFT to monitor student progress against an expected timeline. Students who take much longer to complete training than expected may not be fully engaged in the training or may be having difficulty with the material. These are conditions that might prompt a response by GIFT. Finally, it can play a role in quality control of GIFT courses. For example, if segments of a course take much longer than expected across multiple trainees, GIFT could flag those sections for review by the course author to insure that the material is presented clearly.

Determining the ROI for training is not always easy. As Fletcher and Chatham (2010) put it, how does one determine the benefit of a pound of training? In some cases it may be fairly straight forward. For example, one might measure the increase in revenue produced by the introduction of new training for a sales staff. While this may work for commercial businesses, the military is not a profit making organization, therefore one must look at other factors like cost avoidance to get a measure of ROI.

Determining this can be quite difficult as one rarely has before and after data on the operational impact of training. In rare cases it can be found. For example, Fletcher and Chatham (2010) examined the benefits of Top Gun training given to pilots during the Vietnam war. Because of this training, kill ratios of Navy pilots improved from 2.4 enemy kills per loss up to 12.5 enemy kills per loss. The authors determined that the training had reduced U.S. losses by about 10-12 aircraft during the war When they looked at the cost of procuring and employing that many aircraft during the war, they calculated that the training had saved the Navy about $132 million dollars for an ROI of about 2.5.

Determining the ROI for adaptive vs. non-adaptive training in terms of cost avoidance measures in an operational context would be very difficult. Adaptive training is still relatively new and opportunities to do side-by-side comparisons with traditional non-adaptive training are virtually non-existent. Rather than trying to quantify an impact in the operational environment however, we can look at the impact in a training environment. Specifically, one of the key advantages of adaptive training would be to reduce the overall time needed to deliver the training to a population of trainees.

A challenge for authors of adaptive training is determining how *adaptive* the training should be. While adding adaptive features can potentially save training time, it also increases the cost of development. How does one determine, when the training is adaptive enough? Using an ROI metric can help to answer this question. On one hand is the cost of adding the adaptive feature. On the other hand is the value of the time saved by that adaptive feature. The value of that time could be calculated by looking at the total salary paid to the trainees over that time (e.g., 1,000 trainees/year x .5h/trainee x $35/h = $17,500/year). So, as long as the cost of adding the adaptive feature was less than value of the time saved, there would be a positive ROI and therefore justification for adding that particular adaptive feature.

As can be seen, our model supports this strategy for the design and development of adaptive training in GIFT by helping to predict the effect of adaptive features on the training time for a known population of learners.

There are several challenges we may face as we develop this model. First, the initial MAST skill tree may not contain sufficient variables to predict adaptive training completion times. Our initial literature review and analysis have identified a potential set of most influential variables, but these variables may not be reflective of the completion time upon closer inspection. We will mitigate the identified risk by widening the scope of task models to incorporate more predictive variables if necessary.

Second, while the model predictions may be highly accurate, there is a risk that the system will be too difficult or time consuming to use for some or all of the target populations of instructional designers, course managers, and instructional staff. We mitigate this risk by conducting a requirements analysis early in the effort to closely examine the needs of these user groups and design our system and interfaces to best meet those needs. We will apply human factors and user-centered design and understand the challenges of and methods for developing highly useful and usable decision-aiding tools for practitioners.

Third, while this approach combines state of the art probabilistic approaches and identifies key variables from the literature and past experience, there is a potential that the initial predictions will not sufficiently account for the variability of trainee completion times. We plan to mitigate this risk by incorporating historical data early and adjusting the analysis techniques to capture the maximum amount of variability from data that can be reasonably collected in the field.

When complete, this will be the first system to predict the completion times of GIFT and to enable effective assessments of the ROI that is useful for key design and implementation decisions of an adaptive training system. It includes an innovative application of the procedure skill modeling the MAST skill tree to flexibly represent the adaptive training content for analysis. It is the first application using a probabilistic programming language (i.e., Figaro) to predict completion times for adaptive training technologies, including both unobserved latent variables and temporal factors, such as trainee fatigue, boredom, or flow.

REFERENCES

Army Research Laboratory. (2015). GIFT 2015-1, Generalized Intelligent Framework for Tutoring Release Page. Retrieved from http://www.gifttutoring.org

Dodds P, Fletcher JD. Opportunities for new “smart” learning environments enabled by next generation web capabilities. Journal of Educational Multimedia and Hypermedia. 2004;13:391–404.

Dynarsky M, Agodini R, Heaviside S, Novak T, Carey N, Camuzano L, Sussex W. Effectiveness of reading and mathematics software products: findings from the first student cohort; March Report to Congress 2007. [accessed 2015 May 15]. http://ies.ed.gov/ncee/pdf/20074005.pdf.

Fletcher JD. Evidence for learning from technology-assisted instruction. In: O’Neil HF, Perez R, editors. Technology applications in education: a learning view. Mahwah (NJ): Erlbaum; 2003. p. 79–99.

Fletcher, J. D., and R. E. Chatham. 2010. “Measuring Return on Investment in Military Training and Human Performance.” In Human Performance Enhancements in High-Risk Environments, edited by J. Cohn and P. O’Connor, 106–28. Santa Barbara, CA:

Praeger/ABC-CLIO.

Goldberg, B., & Hoffman, M. (2015). Adaptive course flow and sequencing through the engine for management of adaptive pedagogy (EMAP). In *Proceedings of the AIED Workshop on Developing a Generalized Intelligent Framework for Tutoring (GIFT): Informing Design through a Community of Practice* (pp. 46-53). Madrid, Spain.

Graesser AC, Conley M, Olney A. Intelligent tutoring systems. In: Harris KR, Graham S, Urdan T, editors. APA Educational Psychology Handbook: vol. 3. Applications to Learning and Teaching. Washington (DC): American Psychological Association; 2012. p. 451–473.

McDonnell Douglas Corporation. A survey and analysis of military computer-based training systems: (A two part study). Vol II: A descriptive and predictive model for evaluating instructional systems; 1977. Defense Advanced Research Projects Agency. [accessed 2015 October] http://www.dtic.mil/dtic/tr/fulltext/u2/a043358.pdf.

Merrill, M. D. (1983). Component display theory. In C. M. Reigeluth (Ed.), *Instructional-design theories and models: An overview of their current status* (pp. 282-333). Hillsdale, NJ: Lawrence Erlbaum Associates.

Olney AM, Person NK, Graesser AC. Guru: designing a conversational expert intelligent tutoring system. In: Boonthum-Denecke C, McCarthy P, Lamkin T, editors. Cross-disciplinary advances in applied natural language processing: issues and approaches. Hershey (PA): Information Science Publishing; 2012. p. 156–171.

Pfeffer, A. (2012). Creating and manipulating probabilistic programs with Figaro. Workshop on Statistical Relational Artificial Intelligence (StarAI).

Sottilare R, Brawner KW, Goldberg BS, Holden HK. The generalized intelligent framework for tutoring (GIFT). Orlando (FL): Army Research Laboratory (US); Human Research and Engineering Directorate (HRED); 2012a [accessed 2015 May]. https://gifttutoring.org/attachments/152 /GIFTdescription_0.pdf.

Sottilare R, Goldberg BS, Brawner KW, Holden HK. A modular framework to support the authoring and assessment of adaptive computer-based tutoring systems (CBTS). In: Proceedings of the Interservice/Industry Training Simulation and Education Conference; 2012 Dec 3–6; Orlando, FL. Arlington (VA): National Defense Industrial Association; 2012b.

Sottilare, R. A. (2014). Using learner data to influence performance during adaptive tutoring experiences. In *Proceedings of International Conference on Augmented Cognition--HCII2014* (pp. 265-275). Crete, Greece: Springer.Steenbergen-Hu S, Cooper H. A meta-analysis of the effectiveness of intelligent tutoring systems on K-12 students' mathematical learning. Journal of Educational Psychology. 2013;105(4):970–987.

Steenbergen-Hu S, Cooper H. A meta-analysis of the effectiveness of intelligent tutoring systems on college students' academic learning. Journal of Educational Psychology. 2014;106:331–347.

VanLehn K, Graesser AC, Jackson GT, Jordan P, Olney A, Rosé CP. When are tutorial dialogues more effective than reading? Cognitive Science. 2007;31(1):3–62.

VanLehn K. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. Educational Psychologist. 2011;46 (4):197–221.

ABOUT THE AUTHORS

Dr. Gregory Goodwin *is the acting Branch Chief of the Creative and Effective Technologies Branch at the Army Research Laboratory – Human Research and Engineering Directorate in Orlando, FL. For the last decade, he has worked for the Army researching ways to improve training methods and technologies. He holds a Ph.D. in Psychology from Binghamton University and an M.A. in Psychology from Wake Forest University.*

Dr. Jong Kim *is a postdoctoral fellow at the US Army Research Laboratory, Orlando, FL. Kim received his PhD in Industrial Engineering at Pennsylvania State University. His research interests lie in the area of Cognitive Science and Engineering. Kim is interested in theories of cognition for the development of intelligent systems. Kim developed a skill retention theory (D2P: Declarative to Procedural) that is being applied to implement a series of intelligent tutoring systems for the Navy in collaboration with Penn State and Charles River Analytics.*

Dr. James Niehaus *is a senior scientist at Charles River Analytics. Dr. Niehaus's areas of expertise include artificial intelligence, training systems, and health technology. Dr. Niehaus is currently the principal investigator on multiple efforts developing new training and health technologies. He is currently leading projects to: enhance physical therapy with video games and virtual coaches and investigate the neural, cognitive, and behavioral relationship between implicit learning and intuition. Dr. Niehaus is collaborating on projects to: identify and model critical skills in laparoscopic surgery, develop a training system for combat medic tourniquet application, and develop training for first responder trauma assessment. Previously, he has worked on projects concerning narrative generation, adaptation and individualization of training content, cultural interaction with virtual agents, and cognitive models of discourse comprehension. Dr. Niehaus has a B.S. in computer science from the College of Charleston and a Ph.D. in computer science from North Carolina State University.*

A Data Analytics Framework to Support Training Effectiveness Evaluation

Chalinda Weerasinghe[1], Clarence Dillon[1], Paul Cummings[1], Michael Smith[1], Rodney Long[2]
[1]ICF, [2]Army Research Laboratory

INTRODUCTION

As the Generalized Intelligent Framework for Tutoring (GIFT) framework becomes more pervasive as a research tool, the ability to evaluate student performance will be a critical consideration in its adoption. Having a common method for evaluating learner performance and course data in a simple user-intuitive way will aid course evaluators, instructional designers, and content managers to better assess the effectiveness of a course. In order to produce accurate measurement of learner performance within an intelligent tutoring system, a robust data analytics framework must include the ability to analyze course performance data, learner attributes, and learning strategies. Given the minimal amount of existing intelligent tutoring data, a tool for generating synthetic data would enhance the ability to build models and conduct robust experiments. This data can be used to establish and validate analytic findings, and develop baselines against which to compare observed performance. This paper introduces a data authoring tool and demonstrates its use.

The paper describes research that demonstrated an automated data generation capability that supports course (and performance) evaluation within the GIFT framework. We present a method to generate a simulated class using distributional properties that can be adjusted and visualized by the author. The data generation tool automatically creates dependencies between variables in a logical and consistent manner using a scale-invariant correlation measure (Kendall's tau, τ) from which data can be sampled and replicated. In addition to validating and calibrating the analytics engine, iterative simulations can provide expectations against which observed performance for new or modified GIFT instruction can be compared. The tool has been designed to allow intuitive inputs from users who do not possess strong statistical backgrounds by providing recommended settings and visual accompaniments to aide in data generation. These capabilities will be published to a web-based tool that will run alongside GIFT.

PREVIOUS RESEARCH

"The Changing Face of Military Learning," (Shatz, 2015) points to the complexity and rate of change in Army learning, and emphasizes the need to find new ways to empower learners. They specifically focus on increased investments in expanded set of training competencies, and developing more efficient and agile pathways to expertise. Methods for assessment and evaluation have been explored extensively in the psychology literature, but their application in the context of intelligent tutoring systems represents a less researched topic. The model for the analytical framework draws from foundational research published in Long et al. (2016a) that describes broad directions for analytics research in intelligent tutoring. Long, et al. (2016b) expanded on this research to develop a proof of concept system and user interface. This research explored the use of demographic factor analysis as a tool for performance effectiveness evaluation (Long et al., 2016b). The research successfully evaluated use cases to determine learning outcomes and gains, determinants of success, and recommendations for improvement, but also highlighted the need for a more robust data generation capability (Long et al., 2016b).

The current research builds on existing analytics framework development as described in Long et al. (2016b). Utilizing Basic Rifle Marksmanship as a use case, the research identified key demographic factors that formed the basis for simulated data. This research also introduced the application of the standard Kirkpatrick model for performance evaluation. Widely used in the realm of psychology and instructional systems design, this model posits four levels of evaluation that are used to evaluate performance (Kirkpatrick, 1994). The authoring tool focused on developing data to support what are known as Level 1, which focuses on the learner's reaction and satisfaction post-event, and Level 2, which focuses on the knowledge and skill gains that the learner exhibits.

METHOD

The research team simulated learner data following a process intended to imitate an objective (real-world) Learning Management System, Learning Records Store (LRS), and GIFT environment. An author, which could be an instructor, course designer, or researcher using GIFT, describes one or more learner "personas," which represent the types of learners who notionally populate a class. Persona attributes are user-defined characteristics of learners that broadly describe a group of learners: *e.g.*, "males, ages 18-24 from suburban, middle-class backgrounds and little previous job experience." We expect such data to be available in the objective learning environment: demographic and biographical data from learner profiles and biographical surveys; previous course and test performance from an LRS; and data from the current course, such as assessment scores and operationalized attitude surveys. In addition to variable names and the range of possible values (minimum/maximum or enumerated), authors provide information about distributional parameters (location, scale, and shape), using an interactive graphical tool. The user can specify bivariate dependencies between attributes, and the resultant joint distribution represents a narrative for each persona (see Appendix for screenshots of the authoring tool). Finally, the user specifies the mix of personas and the size of the class population: e.g., "class *XX* has 250 learners, 40% with *persona A* and 60% *persona B.*" The data authoring tool generates a heterogeneous set of learner profiles and learning records with statistical characteristics and dependencies that fit each persona generation scenario (collection of persona narratives).

These data sets form the basis for effectiveness evaluation utilizing the Kirkpatrick Model. The current focus is on Kirkpatrick Level 2, or student learning and knowledge gain. Also referred to as student performance, the most common Level 2 measurements are pre-test and post-test assessments given to the learners. With this simulated data, the factors of age and experience can then be analyzed with respect to these performance assessment outcomes. These measurements form the basis for automating the data analytics framework by providing a tool to assess relative impact on performance that is extensible to many different factors and assessment outcomes. Level 1 data, or learner reaction and satisfaction, is also readily available in GIFT and can be also be easily adapted to this data analytics framework.

Simulating Dependency

The research team utilized copula functions as an efficient method for describing bivariate joint probability distributions with uniform marginals (Joe, 1997; Nelsen, 2006). Sklar's theorem (Sklar, 1959) showed that any multivariate joint distribution can be broken into two components: the univariate marginal distribution functions that come together to make the joint distribution and the copula function that describes the dependency structure between these univariate distributions. The converse is also true: any set of univariate marginal distributions combined with an appropriate copula function will generate a multivariate joint distribution. Moreover, the multivariate joint distribution created is unique if the underlying marginal distributions are continuous. Even if they are not continuous, there is uniqueness over certain ranges of the marginals (Sklar, 1959). This research used the converse of Sklar's theorem to combine variables with arbitrary continuous margins with a copula family that described the desired dependency structure (described below) to form a unique, joint distribution. The tool currently implements the Archimedean class of copulas (Clayton, Frank and Gumbel families) due to their widespread use, flexibility and versatility (Genest & Rivest, 1993; Genest, Nešlehová & Ziegel, 2011).

Use Case

The research team tested data generated by the authoring tool by devising our own persona scenarios. We used methods and attributes developed in prior research (Long et al., 2016a; Long et al., 2016) and attempted to detect these personas and their narratives. The scenario is intentionally simple in an attempt to preclude spurious confirmation. We created *persona A* and *persona B*, each with four attributes: age (between 18-30 years); experience (0-10 years); pre-course assessment (pre-test) score (a proxy for general aptitude, between 0-100); and a predicted post-course assessment score (post-test, also on a 0-100 scale). In this use case we assumed all variables to be continuous.

In our narrative *persona A* learners were on average younger and less experienced, but scored higher on the post-test compared to *persona B* learners, who were older and more experienced, on average. Members in both A and B scored almost identically on the pre-test. Narratives for both *A* and *B* learners included dependencies across some attributes, which we describe next in detail, along with the distributional assumptions used for the simulation, summarized in Table 1.

Table 1. Persona Attributes, Distributions, and Parameters

	Persona A		Persona B	
attribute	*distribution*	*parameters*	*distribution*	*parameters*
age	normal	$\mu = 23,\ \sigma = 1.5$	normal	$\mu = 26,\ \sigma = 1.0$
experience	lognormal	$\mu = 3,\ \sigma = 0.1$	lognormal	$\mu = 4,\ \sigma = 0.1$
pre-test	normal	$\mu = 80,\ \sigma = 5.0$	normal	$\mu = 80,\ \sigma = 5.0$
post-test	normal	$\mu = 86,\ \sigma = 3.0$	normal	$\mu = 80,\ \sigma = 2.0$

Persona A learners' age and experience were positively correlated (moderate to strong, Kendall's $\tau \approx 0.6$), but at lower ages there was lower experience with more variation (more probability of falling at this end), while at higher ages there was higher experience with little variation (less probability of falling at the high end). The pre-test score was independent of age and experience. Instead, their pre-test score was strongly, positively correlated with post-test assessment ($\tau \approx 0.85$) so that those who scored lower on the pre-test scored lower with more variation on the post-test (more probability of falling at the low end), while those who scored higher at the pre-test scored higher in the post-test with little variation (less probability of falling at the high end).

Persona B learners' age and experience were weakly and positively correlated ($\tau \approx 0.3$). But at lower ages, there was lower experience with less variation (less probability of falling at this end) and at higher ages there was higher experience with more variation (higher probability of falling at the high end). The pre-test score was independent of age and experience for *persona B* learners as well. Their pre-test score was again strongly positively correlated with post-test assessment ($\tau \approx 0.8$), with those who scored lower on the pre-test scoring lower with more variation (more probability of falling at this end). Those who scored higher on the pre-test scored higher in the post-test with little variation (less probability of falling at the high end). Figure 1 shows these two personas and their narratives.

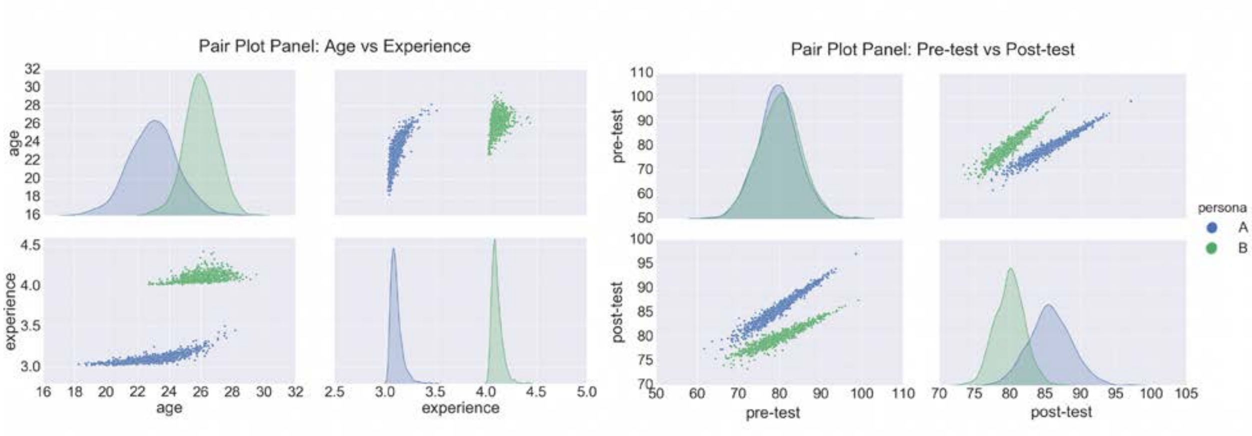


Figure 1. Sample of Attribute Clusters by Persona. From top-left to bottom-right, left to right across: the distributions for age; the scatterplot of experience and age; the distributions for pretest scores; the scatterplot of posttest and pretest scores; the scatterplot of age and experience; the distributions for experience; the scatterplot of pretest and posttest scores; and the distributions for posttest scores.

RESULTS

We present our results based on three levels of analytical validation of the simulated data. First, we review the extent to which the two personas can be recovered through analysis. Next, we validate between-variable relationships. Finally, we validate the parameters within variables, compared to the specifications for simulation. The statistical engine has an overall architecture that uses a three-level analytical design where incoming data is first examined for overall patterns using data mining tools, while uncovering underlying data distributions. At the next level the random variables are analyzed for underlying structures and relationships across variables, and finally within variable data patterns are analyzed.

Level 1 Validation

At the first level, the engine tried to recover the personas that the aggregate data described. The two personas were unique, and using two unsupervised learning cluster methods (k-means and agglomerative hierarchical) the system tried to identify the personas. Originally, the training set had 1000 students each for A and B (the class had 2000 students). Initially, k was specified as 2 (Table 2), and the system classified the observations accurately with an error rate of 0.0075 (15 out of the 2000 students were misidentified), where A had 15 students under-classified (data was standardized). The hierarchical clustering was able to identify the two clusters as well, while noting that one of the clusters could also be broken into two at a lower level. Table 2 also presents the deviations from total averages for the two clusters across the four variables, showing that our clusters represent two very different groups. It also validates persona scenario. The r-squared was moderate (53.3%), but improvements can only be achieved with higher classification error. Currently, code is being developed to locally optimize on the classification and find the best k out of a set of values (k<10) that fits the data if the training set's cluster assignment is known.

Once this was done the engine created histograms/kernel densities for the variables (see Figure 1 above), while generating the empirical distributions. The empirical distributions will also be used for data replication purposes. The densities clearly showed that the generated personas had the variable distributions prescribed to them. Even though groups A and B had distinct unimodal distributions when created, the aggregate data was unimodal only for the pre-test variable, as per the persona generation scenario. The Shapiro-Wilk, Kolmogorov-Smirnov, and Anderson-Darling tests showed that the persona variables followed their specified distributions (not shown).

Table 2: Level 1 Analysis Results

	cluster		error rate
persona group	*A*	*B*	
size	895	1015	0.0075
relative size	0.493	0.508	
R^2			0.533

deviations from total average

persona group	A	B
age	-0.782	0.759
experience	-0.999	0.970
pre-test	-0.015	0.014
post-test	0.767	-0.744

Level 2 Validation

At the second level of analysis, the system tried to recover the between-variable relationships that were built in (Table 3). Kendall's τ tests correctly identified all four bivariate copula correlation values (with repeated simulation to validate these findings) to a very high degree of accuracy (p-values < 0.00001) within the two personas, while also noting that all of the other variables were not correlated with each other ($\tau < 0.03$ for all of them). For the most part, in the aggregate data the class showed dependencies that would have been anticipated given the persona narratives and overall scenario. The correlation between age and experience was $\tau \approx 0.68$, while between pre-test and post-test it was $\tau \approx 0.45$ (p-values $<$ 0.00001). The only other significant result was a moderate negative relationship between experience and post-test ($\tau \approx -0.45$, p-value < 0.00001), which was part of the generation scenario since those in A (lower experienced) did better than those in B (more experienced).

To examine an explanatory model, multiple linear regression was used after validating some of the Gauss-Markov assumptions with post-test as the dependent variable. Table 3 shows model results for the entire class as well separately for the two personas. As expected, the linear model performed very well to the data with high goodness-of-fit measures. The model had low multicollinearity (variance inflation factors were all less than 2.75), which validated the linear structure used in the copula creation code with a random component introduced. All of the coefficients were positive and significant with the exception of experience, which had the largest negative effect on post-test scores (all p-values < 0.00001). The magnitude of the effect was not anticipated but also not surprising given the negative relationship between experience and post-test score, and the fact that experience was following lognormal distributions that were sufficiently far apart (see Table 1). Overall, the regression results validated the persona generation scenario, with all three variables explaining post-test scores well, and the model being a very good fit to the data.

Table 3: Level 2 Analysis Results

	Correlation Results		
attribute pair	Persona A	Persona B	Aggregate Data
age - experience	0.5859*** (0.0211)	0.3253***(0.0211)	0.6757*** (0.0149)
pre-test - post-test	0.8618*** (0.0211)	0.8045***(0.0211)	0.4469*** (0.0149)
experience - pre-test	−0.0173 (0.0211)	0.0231 (0.0211)	0261* (0.0149)
experience - post-test	−0.0209 (0.0211)	0.0273 (0.0211)	−0.4444*** (0.0149)

The Kendall's tau value is given with the standard errors in parentheses, with $^{*}p < 0.1$ and $^{***}p < 0.0001$. Correlations used to build copulas appear in bold type. Not all correlations are presented; only those important to the results.

	Regression Results			
measure	Persona A	Persona B	Aggregate Model	VIF
intercept	41.0953*** (1.5298)	48.6993*** (1.6876)	63.8740*** (0.4006)	
age	0.0319 (0.0226)	0.0221 (0.0225)	0.1573*** (0.0163)	2.5739
pre-test	0.5791*** (0.0043)	0.3815*** (0.0041)	0.4786*** (0.0039)	1.0017
experience	−0.7045 (0.5942)	0.0471 (0.4424)	−6.3875*** (0.0647)	2.7566
num. obs.	1000	1000	2000	
adj. R^2	0.9472	0.8945	0.9484	
std. error	0.6823	0.6666	0.8741	
F statistic	5978.4951 (< 0.0001)	2825.3421 (< 0.0001)	12250.23 (< 0.0001)	

The dependent variable is the post-test score. Coefficients are given with standard errors in parentheses for the variables, with $^{***}p < 0.0001$. All regressions have robust estimates. Parenthesis with the F-statistic indicates the p-value. VIF is variance inflation factor. Standard error is the standard error of the regression.

Level 3 Validation

The final analysis that was done was at the third level, where within variable statistics were examined to see if they followed variable parameters that were specified (Table 4 is for the entire class). The descriptive statistics that were generated showed that at the persona level the variables approximately had the location and scale parameters that were specified (not shown), while at the class level the data fell within the overall scenario bounds and had statistics that were expected (Table 4). Data graphing capabilities (box-plots) visually validated these findings. Overall, the data validation exercise was a success, with the statistical and analytical engine being able to recover the built in data and variable patterns dictated by the scenario modeled.

Table 4: Level 3 Analysis

measure	age	experience	pre-test	post-test
mean	24.5159	3.6106	79.8313	82.8793
std. error	0.0445	0.0112	0.1128	0.0861
median	24.8664	3.7675	79.8941	82.2737
std. deviation	1.9902	0.5018	5.0424	3.8483
skewness	-0.5890	-1.9553	0.1899	-0.4358
range	11	1	37	24
maximum	30	4	99	97
minimum	18	3	62	73
IQR	3.0699	0.9989	6.4993	5.7788
count	2000	2000	2000	2000

Results

DISCUSSION

The results indicated that the automated analysis tool for GIFT successfully detected the persona groups from the learner population at each level of validation. Based on a number of measures these results were consistent and demonstrated statistical significance. Given the success of this relatively simple model, the data authoring tool can be used to develop more complex interdependencies for larger numbers of variables. Copulas have been demonstrated to be a highly scalable tool for modeling relationships with a number of desirable statistical properties. These results can also be scaled easily to apply to tens or even hundreds of thousands of students if operated on the right system architecture.

Clustering is a flexible tool that can be automated in a deployed environment to automatically categorize real-world data. This will allow insights to be delivered in a more seamless fashion by grouping outputs and students into groups that make sense intuitively. For example, an analytic output of the system could be that experienced students are performing poorly compared to inexperienced students and indicating lower levels of satisfaction. While this result is interesting in and of itself, an automated analytic tool-kit will need to define and identify these groupings and apply them to real world students, a capability demonstrated in the data authoring tool.

The data authoring tool and the automated performance analysis tool are both web-based applications, designed to support learning in a GIFT environment. The results can be used immediately in a GIFT environment to develop notional course performance data for authoring. The results can be used to implement a persona-based, baseline data authoring system. The models can be deployed within any Python environment and push data to a web-based interface. The data authoring tool and analytics engine could both be more useful if GIFT had a native capability to export course-level learner data that includes learner performance evaluations (scored and graded assessments). This would make real-time comparison of learner performance against subgroup baselines possible.

Results form the basis for a reusable testbed for learning analytics research. Results can be used in any environment with assessment data prior to employment. Simulation can be utilized post-employment as a kind of virtual laboratory to generate best case scenarios and isolate specific factors without the noise and messiness of empirical data. These outputs can then be used to calibrate analytics deployed in a number of different settings.

CONCLUSIONS AND FUTURE RESEARCH

Success here forms the basis for implementing and validating a robust, automated analytics capability for GIFT, research that is underway at the time of publication. By hypothesizing, identifying, and validating statistical relationships in a laboratory like setting, the data authoring tool has blazed a clear path to creating reproducible analytical results using empirical GIFT course data. By developing synthetic benchmark data, researchers and experimenters will be able to test their own theories and analytic applications in the GIFT framework with a tool fit for the intended environment. Once completed, the data authoring tool can be used in conjunction with the analytics engine to develop a continuous feedback loop to improve performance.

While initial development focuses on descriptive and inferential analysis, future extensions will easily transition to predictive applications. By developing and validating statistical relationships across increasingly complex sets of variables and interactions, the analytics engine can develop sophisticated learner models that can be evaluated with real-world GIFT interaction data. Once validated, the calibrated models can be used to rank success factors and identify problem areas earlier in the instructional process. For example, a notional data set depicting a relationship between education and performance could be developed in the authoring tool. Real world data could be used to validate this relationship, and predictive measures could be employed to change instructional strategies for these students. The benefit of an authoring environment is that while these relationships will play out across many variables over time, the individual effects can be isolated, studied, and ultimately employed to improve instruction.

Even though only bivariate copulas were used in the simulation, code is being developed to use multivariate copulas (using vines) in more general dependency settings (Bedford & Cooke, 2002; Kurowicka & Joe, 2011). Even though the code does not currently permit recovery of the underlying copula structures from the data, while only being able to obtain the bivariate dependencies that were built in, the vine code that is being developed will be able to identify multivariate copula structures in addition to identifying dependency strength.

Finally, while the initial demographic analysis variables are limited, it is straightforward to scale up and include more variables and inputs in the data authoring tool. Extending the simulation to include competency modeling and concept mapping is one such natural extension of this capability. In addition, granular event level data such as user activity in the Event Reporting Tool could yield additional predictive insights, as has been highlighted in other learning analytics research such as Chatti et al. (2012).

REFERENCES

Bedford, T.J. & Cooke, R.M. (2002). Vines- A new graphical model for dependent random variables. *Annals of Statistics 30*(4): 1031–1068.

Chatti, M.A, Dyckhoff, A.L., Schroeder, U., & Thüs, H. A (2012). Reference Model for Learning Analytics. *International Journal of Technology Enhanced Learning 4*(5/6), pp. 318-331.

Genest, C., J. Nešlehová, & J. Ziegel (2011). Inference in multivariate Archimedean copula models. *Test 20*(2), 223–256.

Genest, C. & L.-P. Rivest (1993). Statistical inference procedures for bivariate Archimedean copulas. *Journal of the American Statistical Association 88*(423), 1034–1043.

Joe, H. (1997). *Multivariate Models and Dependence Concepts.* London: Chapman & Hall.

Kirkpatrick, D. L. (1994). *Evaluating training programs: The four levels.* San Francisco: Berrett-Koehler.

Kurowicka, D. & Joe, H. (2011). *Dependence modeling: Vine copula handbook.* World Scientific, Singapore.

Long, R., Smith, M., Dass, S., Dillon, C., & Hill, K. (2016a). *Data Analytics: Techniques and Applications to Transform Army Learning*. ModSim World Conference, NDIA, 2016.

Long, R., Smith, M., Dass, S., Dillon, C., & Silverman, J. (2016b). *Authoring Tools to Assess Training Effectiveness in Intelligent Tutors.* I/ITSEC Conference, NDIA, 2016.

Nelsen, R. B. (2006). *An Introduction to Copulas.* New York: Springer. Chapter 4.

Shatz, S. (2015). *The Changing Face of Military Learning*. Interservice/Industry Training, Simulation, and Education Conference

Sklar, A. (1959). Fonctions de r´epartition ´a n dimensions et leurs marges. Publ. Inst. Statist. Univ. Paris, 8, 229-231.

ABOUT THE AUTHORS

***Chalinda Weerasinghe** is a research consultant who specializes in International Relations, Econometrics, Political Economy, and Development Economics. He undertook doctoral studies in Economics and Government and Politics at the University of Maryland, College Park with specializations in econometrics, international relations and development economics. He has an MS in Economics and an MS in International Relations from the Georgia Institute of Technology and a B.S. in Mathematics, Economics and History and Political Science (triple major) from Shorter University.*

***Clarence Dillon** is a computational social scientist and Consultant at ICF. He authored the ontology for the Department of Defense strategic planning scenarios and established the first collaborative, semantic web platform used in DoD. Mr. Dillon holds a Bachelor of Arts in International Affairs, a Masters of Social Science in International Relations and he conducts graduate research in social complexity at George Mason University.*

***Paul Cummings** is Senior Fellow in the Center for Advanced Learning Systems at ICF. He has over 19 years of technical and management leadership experience in the education, simulation, and training community. Mr. Cummings currently develops immersive technology systems for a wide range of clients. Mr. Cummings is a PhD student in the Krasnow Institute Computational Social Science program.*

***Mike Smith** is a Senior Technical Specialist with ICF and has over 13 years of experience in data analytics, strategic planning, and management. Mr. Smith advises government clients on how to adapt analytics practices to improve their organizational performance. Mr. Smith has a B.A. in International Economics from Longwood University and a Master of Public Policy from Georgetown University.*

***Rodney Long** is a Science and Technology Manager at the Army Research Laboratory in Florida and is currently conducting research in adaptive training technologies. Mr. Long has a wide range of simulation and training experience spanning 29 years in the Department of Defense and has a Bachelor's Degree in Computer Engineering from the University of South Carolina and Master's degree in Industrial Engineering from the University of Central Florida.*

APPENDIX

This appendix includes screen-capture images showing the user interface for creating Personas and identifying correlations between Persona attribute variables.

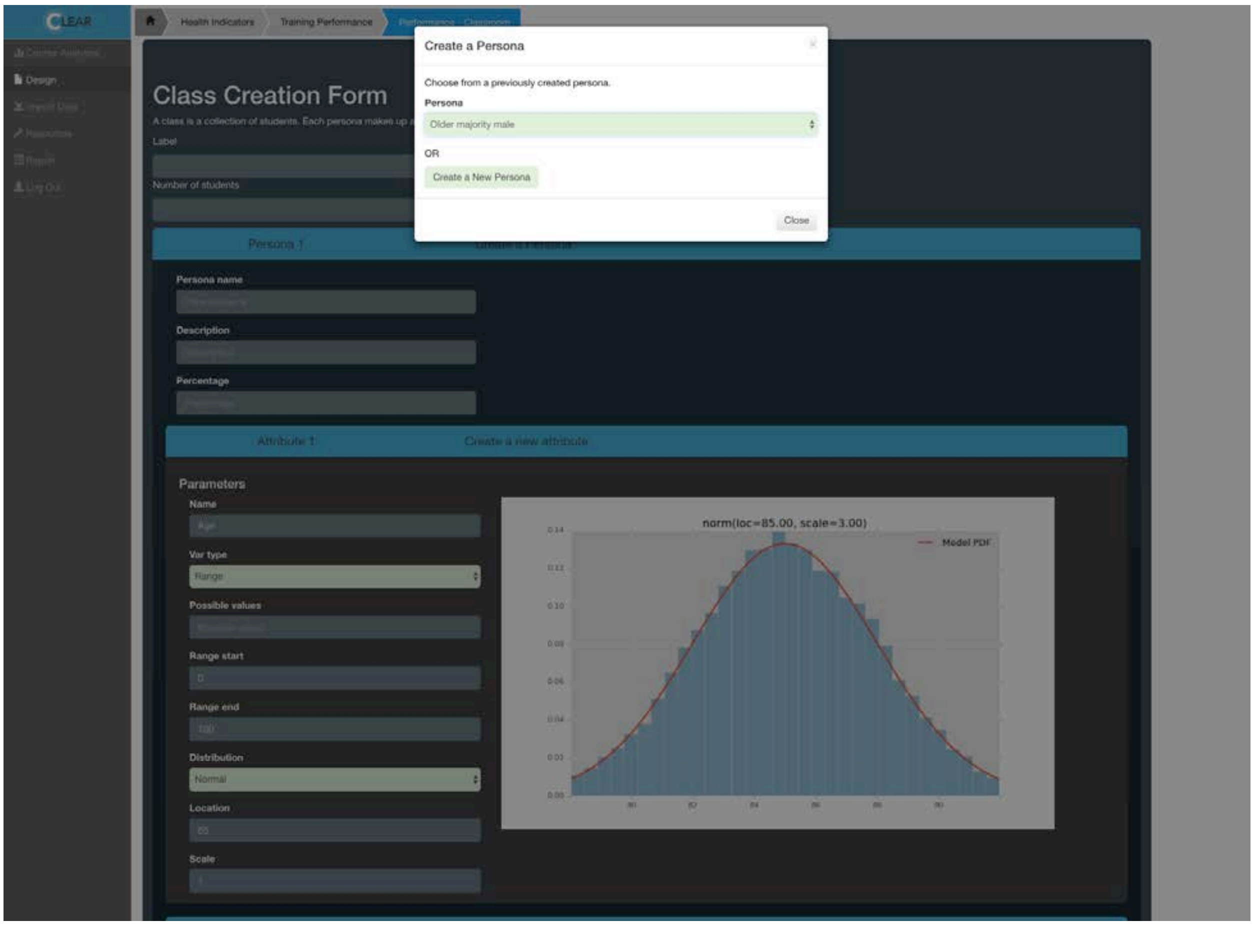


Figure A-1. From the Class Creation Form, a user can design a persona and visualize the patterns of each attribute by itself. Form options are intended to be non-technical and to guide a user through the process in preparation to designing correlations in the next form. The user types a name for the attribute and uses a selection box to choose the type of variable—whether it is from the learner profile, the LRS, a biographical survey, or the current course. The user can choose if the attribute is a discrete variable with enumerated options or a continuous variable with minimum and maximum values over a range. The user can select the shape of the distribution from several options (normal, half-normal, lognormal, gamma and Chi-squared). After selecting the shape of the distribution, they select the location (mean) and scale (standard deviation).

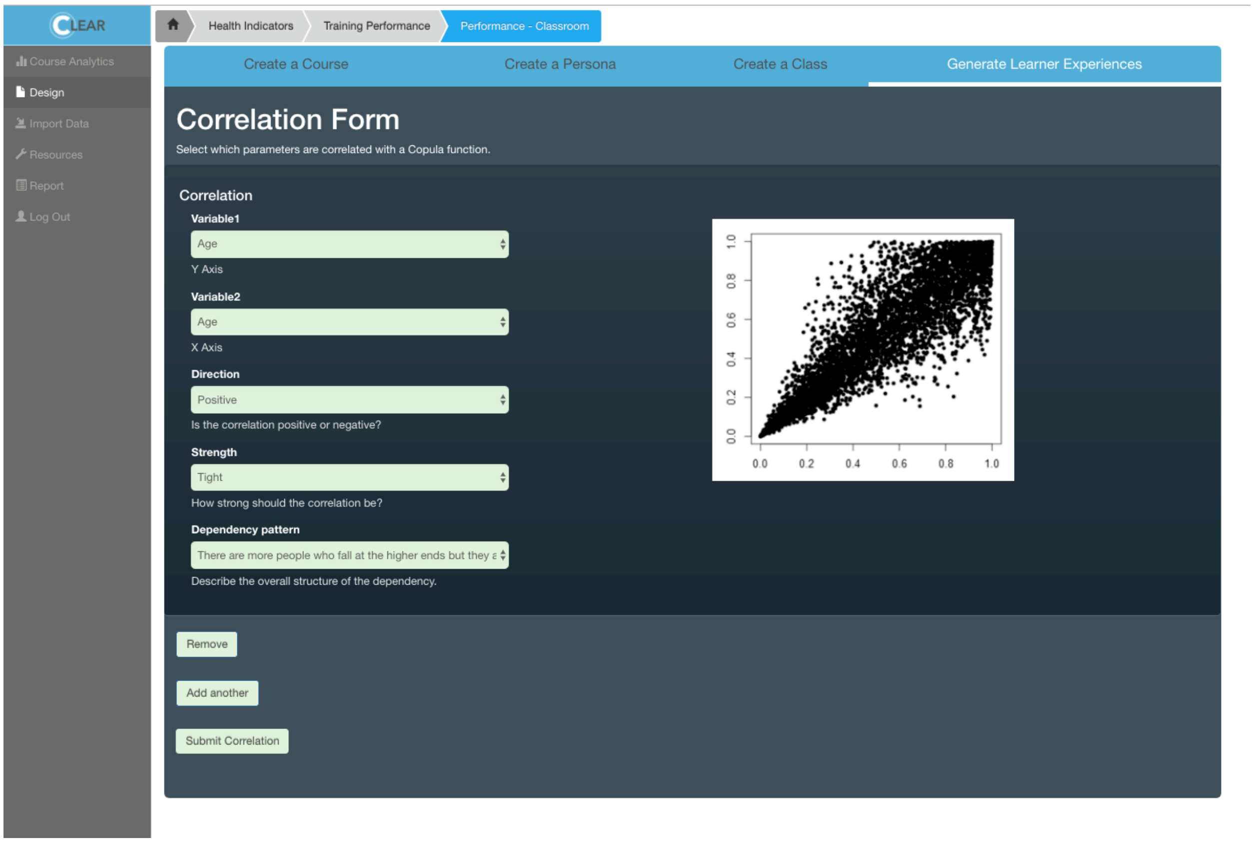


Figure A-2. The Correlation Form guides a user through designing joint distributions between any two persona attributes. Options include the direction of correlation (positive or negative), the strength of the correlation (tight, moderate or loose) and the general pattern (tighter at the bottom and the top, tighter at the bottom and looser at the top, or looser at the bottom and tighter at the top). In the background, the application associates these descriptions with Frank, Clayton and Gumbel copula families, respectively. The plot to the right is visual example of the correlation generated by the form field parameters.

Educational Data Mining Using GIFT Cloud

Alan Carlin[1], Evan Oster[1], Chris Nucci[1], Diane Kramer[1], Keith Brawner[2]
Aptima, Inc.[1], Army Research Laboratory[2]

INTRODUCTION

Intelligent Tutoring Systems (ITS), like human instructors, make frequent decisions about what to present to the student. These decisions include what courses or content to present next, as well as what type of After Action Review (AAR) to present to the student after each course. Ideally, the AAR would be Adaptive (AAAR). In this work, we analyze the decisions of course content and presentation. We construct a student model which models the skills necessary, the effectiveness of each course at training each skill, and the relationship between in-scenario measures and student skill-level. If the student model is accurate and represented mathematically, then decision-theory can be used by the ITS to select courses and course content.

There are two ways to develop the mathematical model of skills, effectiveness, and transition. A first way is that a Subject Matter Expert (SME) or an instructor can carefully build it, using an interface that translates SME intuition to model parameters. In this work, however, we explore tools to facilitate a second option, that of building the model automatically from a corpus of student data. We report on progress towards an enhanced version of the Newtonian Talk tutor (Zhao et al., 2015), on GIFT Cloud, using these mathematical modeling methods. Enhancements include the ability to select an AAR adaptively, the ability to display that AAR, and the ability to sequence courses in a customized order. Each of these requires an ability to learn information about the training domain. To this end, we report on a data collection study which will produce the information necessary to build the enhanced tutor.

DATA COLLECTION METHOD

First, a mathematical mapping of skills and effectiveness was created from data. In order to facilitate the modeling process, data was collected on 44 subjects using the GIFT-Powered NewtonianTalk tutor (Zhao et al., 2015). Prior to the experiment, participants were asked to sign an informed consent form, complete a brief survey, and take a pretest. The survey gathered demographic information such as age, education level, gender, and average physics grade. The pretest consisted of 8 questions based on figures and gathered data on participant's physics knowledge. An example is shown at the left of Figure 1. Once the survey and pretest were completed, the data collection began.

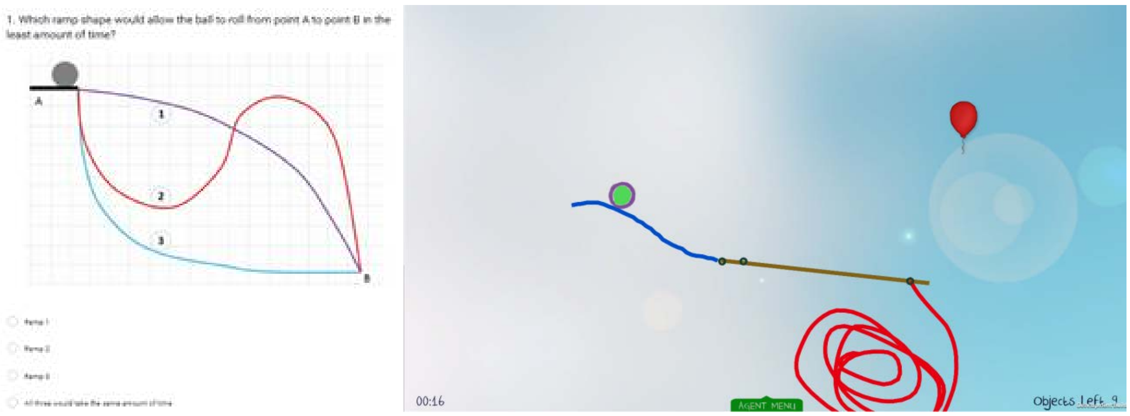


Figure 1: (Left) Pre-test question within Newtonian Talk tutor. (Right) Courses within Newtonian Talk.

During data collection, participants completed a series of physics puzzles in Newton's Playground (an example is shown at the right of Figure 1). This is a game that presents learners with puzzles to solve by strategically creating physics-based objects in a 2D virtual space in order to manipulate a ball, which pops a balloon. Figure 1 shows the ball in green, the red balloon in the upper right, and a series of student-drawn objects in both blue and red.

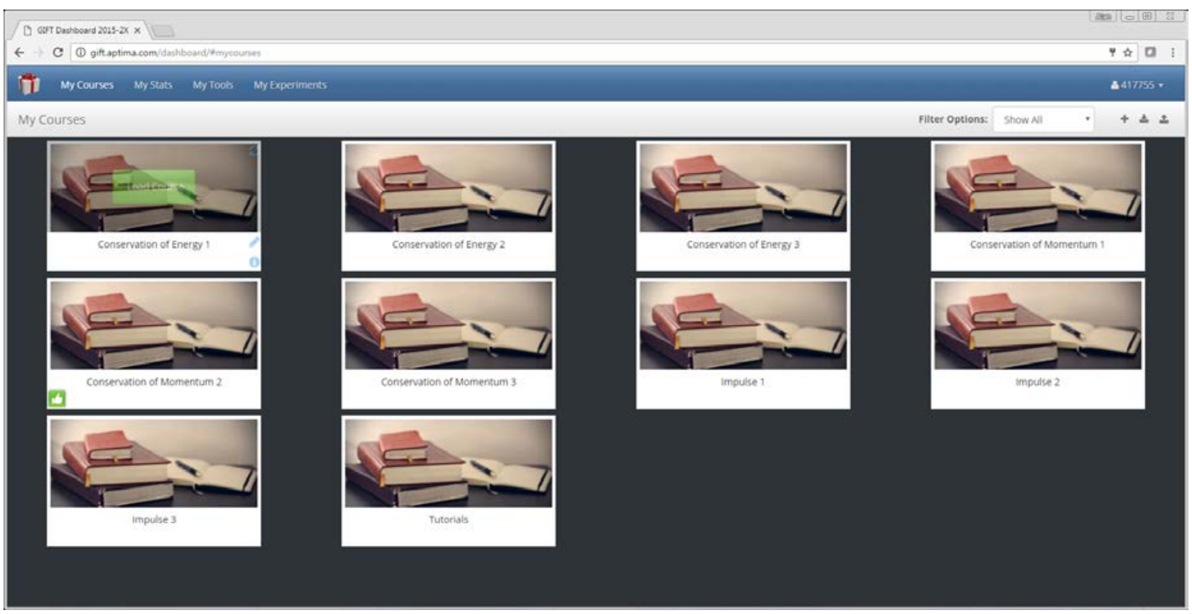


Figure 2: Selection screen that includes a Tutorial and 9 Newtonian Talk Courses.

Participants were randomly assigned a unique user ID which designated a unique path through the series of 9 such puzzles. As part of the reported effort, we modified the GIFT software so that one particular course was recommended at random (shown by the green thumb on the leftmost course in the second row of Figure 2). Subjects were instructed to simply always select the recommended course.

At the end of the session, participants completed an 8 question posttest similar in nature to the pretest to gather data as a comparison point of physics knowledge to the pretest. This was followed by a short debriefing where they learned more about the purpose of this data collection. The data collection lasted about 45 minutes to 1 hour, but the exact amount of time depended upon learning pace. Learners were able to take breaks at any time during the session.

IMPACT ON GIFT FRAMEWORK

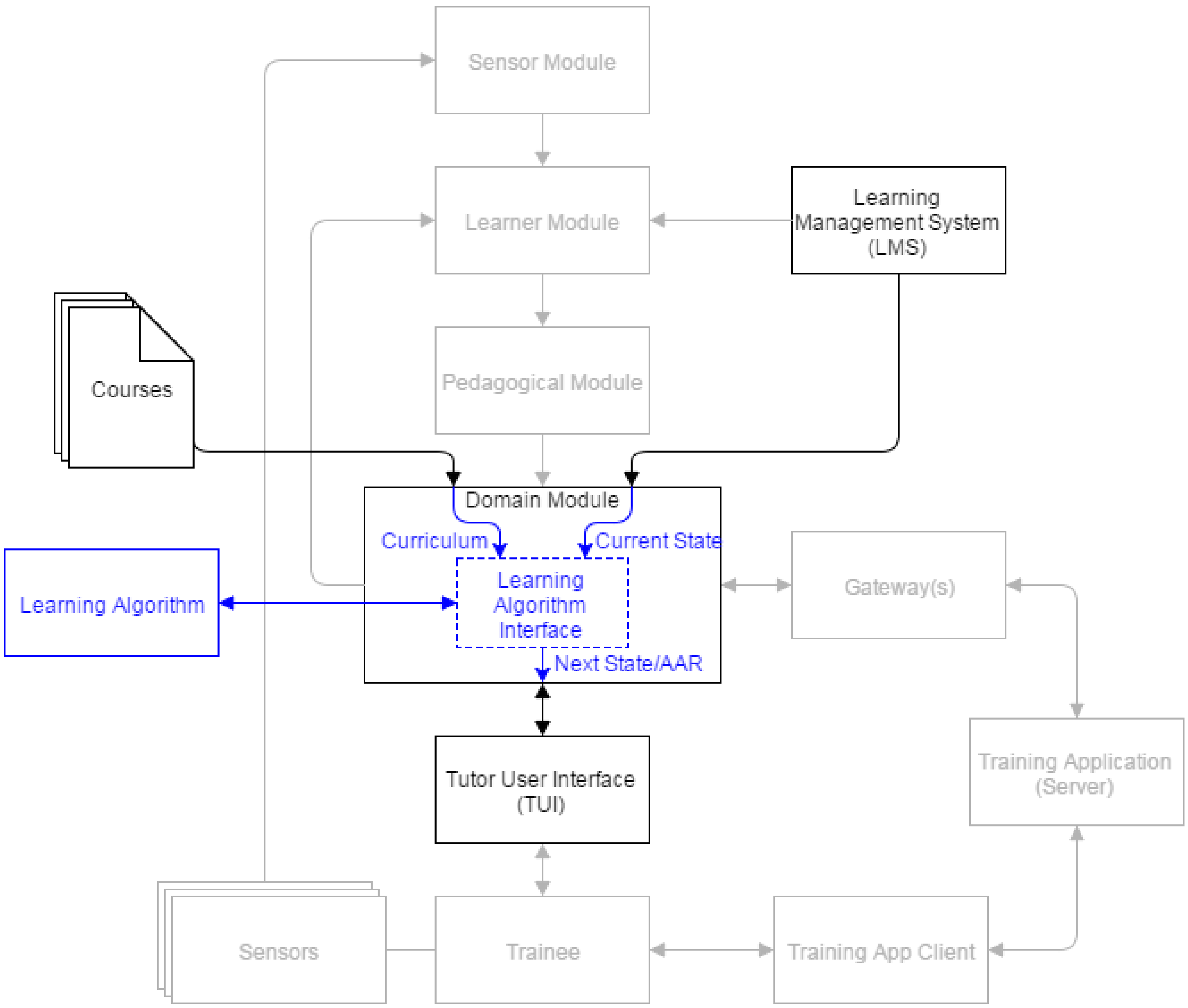


Figure 3: Modules affected within the GIFT framework.

Support for the data collection entailed several modifications to the GIFT framework. Figure 3 highlights the modules affected by the data collection. The workflow through the framework is as follows:

1. The student takes a pre-test to assess physics knowledge.
2. An adaptive learning algorithm (e.g., MDP) determines the next learner course.
3. An AAR screen shows the results of the learning algorithm.
4. The learning algorithm also makes a recommendation the students' next course.
5. The student takes the next course.
6. The process repeats (go to step 2 above) 9 times.
7. After the 9th scenario, the learning loop ceases and the student takes a post-test.

The contributions of this framework lie in Steps 2, 3, and 4. To support the data collection and a follow-up effort, we made modifications to GIFT software and Newtonian Talk so that course content and AAR (an AAAR) was personalized to the individual. These modifications included:

- The Learning Management System (**LMS)** was modified to store a custom construct called student state, described further in the next section. Related to the LMS modification, messages were added throughout the system so that state could be transmitted between modules.
- The **Domain Module** was enhanced to include classes that represent an adaptive policy, as well as the logic for utilizing the adaptive policy and using the policy to return AAR information. The Domain Module was also used to get/set user state in the LMS.
- **Tutor UI**: The tutor UI was tweaked to display custom AAR information.

These modifications all represented improvements on previous work towards customizing GIFT (Hruska et al., 2011). In the next section, we discuss the State data structure used by both the LMS and Domain Module. The modifications to the Tutor UI will be discussed later in this document.

DATA MINING MODULE

To model the data, we used a Partially Observable Markov Decision Process (POMDP; Smallwood & Sondik, 1973). The POMDP model contains various parameters which must be identified for the specific domain: State, Actions, an Observation function (measures), Transitions, and Reward. Since Reward is usually assigned by a human and reflects the individual instructor's priorities, we will not discuss it further in this document. The other parameters are data mined by the AAR system, and are described below.

State

Definition 1 (State): State is defined as a k-tuple of numbers, with k representing the number of skills in the training domain. For the preliminary Newtonian Talk study, we automatically extracted the course measures from the LMS, and we considered each measure available to the system to measure its own skill. In future studies, this requirement will be relaxed so that each measure does not necessarily need to measure one skill. An example of a state is $< 3,5 >$, referring to a skill level of "3" on the first skill and "5" on the second. The Newtonian Talk domain, since it had 10 measures, had 10 skills. Symbolically, we represent state with the symbol θ.

Definition 1a (State Probability): We refer to the probability of being in a state with the notation Pr(). E.g., Pr(<1,1>) represents probability of being in state <1,1>. We may also denote a given point in time when the student was in that state. That is, $Pr(< 1,1 >^{\{t=0\}})$ refers to the probability that a student was in state <1,1> at time zero.

Actions

The set of Actions was identified as the set of 9 courses in the Newtonian Talk tutor. For each course we associated an id, and we created variables to represent the applicability and difficulty of the course to each component of state. The variable d_i^k was used to refer to the difficulty of course i with respect to skill k, and the weight w_i^k was used to refer to the applicability weight of course i with respect to skill k.

Observation Function

To fit the model, we used Item Response Theory (IRT; Lord, 1980) to fit the observation parameter. In Newtonian Talk, each course was either passed or failed, and we expressed the probability that a student would pass a course as $Pr(correct)$. IRT models performance on items using logistic regression.

Equation 1 (IRT): $$p(correct) = \frac{1}{1+e^{(d_i-\theta)}}$$

Thus, in IRT, the probability of correct completion of a course depends on the course difficulty and the student state. If the course difficulty exceeds the student state, the student is unlikely to complete the course correctly. Conversely, if the student state exceeds the course difficulty, then the probability of completion is high.

The AAAR framework extends the notation by vectorizing it to account for many skills. Let k identify a skill. We modify Equation 1 so that the overall capability of the student to perform on the item, is the sum of the capabilities on the individual skills. This yields:

Equation 2: $$p(correct) = \frac{1}{1+e^{\sum_k w_i^k (d_i^k - \theta_j^k)}}$$

Where d_k^i, and w_i^khave been introduced above, and where θ_j^krepresents the skill level of student j on skill k. Equation 2 differs from Equation 1 in that probability of completion is now a linear weighted combination of difficulties and skill levels. When we want to discuss all skills of a student, we will use a vector, so we would represent all skills of student j with a bar to represent a vector, as in $\overline{\theta_j}$, or an alternative is boldface, $\boldsymbol{\theta_j}$.

Example 1: Suppose a student is at level 3 for skill 1, and level 5 for skill 2. We summarize this by saying $\boldsymbol{\theta} =< 3,5 >$. Suppose item 22 is at difficulty level 6 for skill 1, and 2 for skill 2. That is, $d_{22} =< 6,2 >$. Plugging back into Equation 2, the student is modeled as 50% likely to get the item correct.

Transition Function

We would like to model students that improve as they train, following on the literature of deliberate practice (Ericsson et al., 1993). In the above model, there is only one student variable for each skill θ_j^k, instead we would like to break this out into several variables $\theta_j^{k,t}$ representing the skill level at skill k, by student j, at time t. Our model does not use a specific time like 53.45151 seconds, but rather discretizes into time steps. In Newtonian Talk, t represents the number of courses completed by the student thus far. That is we model student skill after 0 courses, after 1 course, after 2 courses, etc.

We can then model a *transition function*, which we denote as *T*, and represents the probability of student improvement. The transition function takes the form:

Equation 3: $$T\left(\theta_j^{k,t+1} \middle| \theta_j^{k,t}, \alpha\right)$$

This represents the probability of that student j achieving a skill level on skill k the next step (that is, at time $t+1$), given that student's skill level at the current time step (represented by time t), and the training action (i.e., the course) α.

For the current AAR model, we assume transitions are independent between skills. This eventually will not need to be the case, and if transitions were not independent we would use:

Equation 4: $$T\left(\theta_j^{t+1} \middle| \theta_j^t, \alpha\right)$$

We propose two methods to assign this probability. The simpler method is to directly interpret transition probability as an artifact of item difficulty levels and student states through a rule: the closer a course's difficulty to a student's skill level, the more likely the course is to train the student. This corresponds to Vygotsky's Zone of Proximal Development (Vygotsky, 1978). The second is to solve for these probabilities directly based on machine learning the value-assignments for all of the other variables and counting the number of transitions. In this study, we explore this second option.

Goal of Data Mining Study

The purpose of the data mining study was to learn values for all difficulty variables d, all weights w, all transition probabilities $T\left(\theta_j^{t+1} \middle| \theta_j^t, \alpha\right)$ for the Newtonian Talk tutor. Variable values were learned by a Gibbs Sampling algorithm (Geman & Geman, 1984). Values for subject knowledge states θ_j^{t+1}were also learned for the subjects of the data collection study. Knowledge of the domain variable values will allow for the construction of adaptive training algorithms in future studies. The adaptive training algorithms will optimize course selection based on these learned variable assignments.

PRELIMINARY RESULTS

The data collection was completed in February 2017. In this paper, we report on a preliminary analysis.

Data Mining Result

We used domain information from Newton's Playground as well as Gibbs sampling to sample values for the variables discussed in the above section. To facilitate, we defined measures and skills synonymously (i.e., each measure observes a single unique skill). If a measure/skill was present in a course, the course was assigned a "1" for presence of that skill. A summary of activities and puzzles is shown in Figure 4. Figure 4 shows which courses are available, the subjects that they intend to teach, and the activities required to complete them.

Course	Name	draw pin	draw spring	draw anything	draw pinned	draw freeform	draw ramp
Playground 1, Puzzle 1	Tutorial 1						
Playground 1, Puzzle 2	Tutorial 2						
Playground 1, Puzzle 3	Tutorial 3						
Playground 1, Puzzle 4	Tutorial 4						
Playground 1, Puzzle 5	Tutorial 5						
Playground 2, Puzzle 1	Impulse 1	X	X				
Playground 2, Puzzle 2	Impulse 2	X	X				
Playground 2, Puzzle 3	Impulse 3	X	X				
Playground 3, Puzzle 1	Momentum 1			X	X		
Playground 3, Puzzle 2	Momentum 2			X	X		
Playground 3, Puzzle 3	Momentum 3			X	X		
Playground 4, Puzzle 1	Energy 1					X	X
Playground 4, Puzzle 2	Energy 2					X	X
Playground 4, Puzzle 3	Energy 3					X	X

Figure 4: Summary of activities and puzzles. For each puzzle (rows), the related skills/measures are denoted.

After all skills were assigned as present/not present, the course was normalized so that the sum of its applicability variables was 1. Based on the data, skill level was assessed for each of the subjects in the data collection on a 1-10 scale. Below, the assessment is shown for the first several courses of a typical student. Each row of the table represents assessed student state on the various skills, after the r-th course, where r is the row number in the leftmost column.

Course #	General	Draw Anything	Draw Freeform	Draw Pin	Draw Pinned	Draw Ramp	Draw Spring	User Ramp	User Spring
0	2	1	1	2	2	0	0	0	1
1	3	1	1	3	3	0	0	0	1
2	3	2	2	3	3	1	1	0	2
4	3	3	2	5	4	1	1	0	2
5	7	3	2	5	5	2	2	1	2
6	7	4	3	6	5	2	2	2	3
7	8	5	4	7	5	2	2	3	3

Figure 5: Assessed state for one of the subjects of the data collection study. Subject skill level progressed after each course (row).

Overall course difficulties were estimated (on a 1-10 scale) based on sampled fit to the item response equation. Below shows the last sample taken. In future analysis, the average of the samples will be retained. As an example, in Figure 6 the course "p2p1.course.xml" was assigned a difficulty level of 7 as a result of the Gibbs sampling learning process.

Course ID	Sampled Difficulty
p2p1.course.xml	7
p2p2.course.xml	6
p2p3.course.xml	6
p3p1.course.xml	4
p3p2.course.xml	5
p3p3.course.xml	4
p4p1.course.xml	1
p4p2.course.xml	5
p4p3.course.xml	5
tutorials.course.xml	1

Figure 6: Assessed difficulty of various Newtonian Talk courses on a 1-10 scale.

Transition probabilities were found as well, although these values were too numerous to list in a table. The transition data was 4-dimensional: each course, skill, and pre-course state, each post-course state was assigned a probability. The table below shows the probability of attaining various skill levels for the p2p1.course.xml, when the student was at a skill level of 0 before the scenario.

0	1	2	3	4	5	6	7	8	9
71.1%	28.3%	.6%	0	0	0	0	0	0	0

Figure 7: Transition probability for one course, from a starting skill level of zero. Each column represents probability of attaining a new skill level as a result of the course. Reference Figure 50 for the mappings between skills and actions.

Simulated Student Result

Using the learned model, it is possible to simulate students using desired instructional strategies. For example, Figure 8 compares simulated student progress using an adaptive training strategy that intelligently selects NewtonianTalk puzzles, versus a strategy that selects random puzzles. To generate this figure, 10,000 students were modeled by the POMDP produced by the data mining procedure discussed in this paper. Each student transitioned randomly to a new state after each scenario, according to a distribution governed by the POMDP transition function. The adaptive strategy selected the best available puzzle for that given student, whereas the random strategy selected a random puzzle. Figure 8 represents average skill level attained, across all skills (see columns of Figure 4), across a 10000 students.

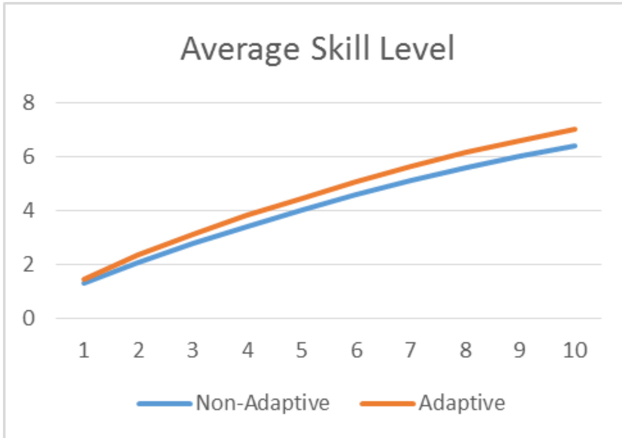


Figure 8: Simulated student skill level using an adaptive training strategy versus using a random one. The x-axis represents number of simulated puzzles, y-axis represents student skill level on a 1-10 scale, as produced by the model.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

In this paper we have reported on a data collection study. The study modified the Newtonian Talk branch of GIFT to sequence courses, to store results in the LMS, and to interpret information about the courses using data mining techniques. A preliminary analysis of the data is described in this paper. With the data collection complete, there are several future directions which will all take place over the next year.

1. Refine and enhance the analysis of the variables described in this study.

2. Use the resulting variable values to parameterize an adaptive training algorithm, and use this algorithm to sequence subjects in Newtonian Talk for a future study, thus proving the efficacy of the GIFT framework on adaptive training.

3. Use student assessments to present an After Action Review. The mockups below show examples of what this After Action Review will look like for future experiments.

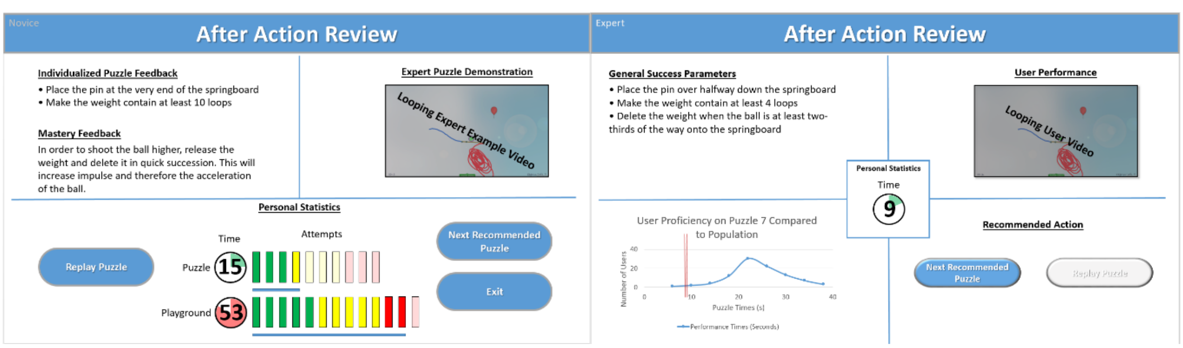


Figure 9: Different AARs are pictured for different users, based on experiences and data mined policy.

REFERENCES

Ericsson, K. A., Krampe, R. T., & Tesch-Römer, C. (1993). The role of deliberate practice in the acquisition of expert performance. Psychological review, 100(3), 363.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. Pattern Analysis and Machine Intelligence, IEEE Transactions on, (6), 721-741.

Hruska, M., Dewey, M., Paonessa, G., Paonessa, M., Poeppelman, T., Nucci, C., & Ayers, J. (2013). Software Design Document (SDD) for Soldier Performance Planner Application. Report to US Army RDECOM Army Research Laboratory – STTC. Problem Solutions.

Lord, F.M. (1980). Applications of item response theory to practical testing problems. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Smallwood, R. D. & Sondik, E. J. (1973). The optimal control of partially observable Markov processes over a finite horizon. *Operations Research, 21*, 1071-1088.

Sottilare, R., Holden, H., Goldberg, B., & Brawner, K. (2013). The Generalized Intelligent Framework for Tutoring (GIFT). In Best, C., Galanis, G., Kerry, J. and Sottilare, R. (Eds.) *Fundamental Issues in Defence Simulation & Training*. Ashgate Publishing.

Vygotsky, L. (1978). Interaction between learning and development. Readings on the development of children, 23(3), 34-41.

Zhao, W., M. Ventura, et al. (2015). GIFT-Powered NewtonianTalk. Generalized Intelli-gent Framework for Tutoring (GIFT) Users Symposium (GIFTSym3), Orlando, FL.

ABOUT THE AUTHORS

Alan Carlin, PhD *is a Senior Research Engineer at Aptima, Inc. His interests focus on problems of artificial intelligence and machine learning. These include problems of decision making under uncertainty, communication between members of a team, and meta-reasoning for decision-makers. His publications include works on Decentralized Partially Observable Markov Decision Processes, game theory, and distributed meta-reasoning in uncertain environments. While at Aptima, he has developed intelligent training systems, NextGen pilot alerting systems, and data mining systems. Dr. Carlin received a Ph.D. in Computer Science from the University of Massachusetts, an M.S. in Computer Science from Tufts University, and a dual B.A. in Computer Science and Psychology from Cornell University. As part of his M.S., he also completed the MIT Lincoln Scholar's Program, sponsored by the Massachusetts Institute of Technology.*

Evan Oster *is an Associate Scientist at Aptima, Inc. with experience in courseware development, game-based training, and augmented reality-based instruction. His current work centers on using cognitive science and instructional design to increase training effectiveness in live and synthetic military environments. He has experience managing interdisciplinary teams responsible for the development and integration of various training products. Past work has involved designing game and simulation-based training using immersive virtual environments for the Navy. Mr. Oster has also created several augmented reality games with a university games lab while providing software development evaluation. Evan Oster holds an M.A. in Instructional Design and Technology and a B.S. in Human Development from Virginia Polytechnic Institute and State University as well as an M.S. in Curriculum and Instruction from Radford University.*

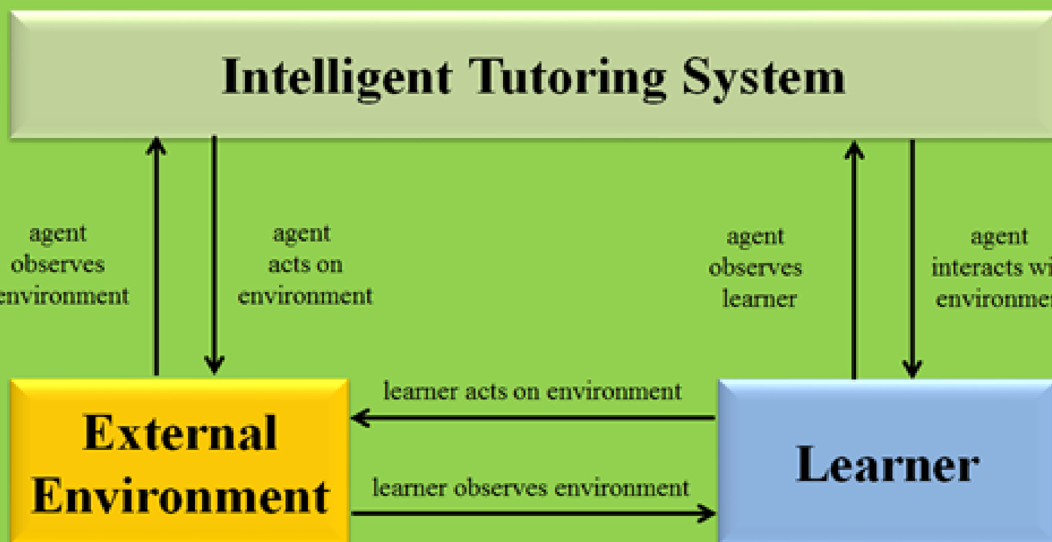
Keith Brawner, PhD *is a researcher for the U. S. Army Research Laboratory's Human Research & Engineering Directorate (ARL-HRED), and is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT). He has 11 years of experience within U.S. Army and Navy acquisition, development, and research agencies. He holds a Masters and PhD degree in Computer Engineering with a focus on Intelligent Systems and Machine Learning from the University of Central Florida. His current research is in ITS architectures and cognitive architectures. He manages research in adaptive training, semi/fully automated user tools for adaptive training content, and architectural programs towards next-generation training.*

Diane Kramer *is a Principal Software Engineer and Certified Scrum Master at Aptima, Inc. with over 25 years of experience designing and developing software applications using various programming languages and platforms. Additionally she has academic experience teaching Computer Science at both the college and high school levels. Her current work at Aptima involves leading small teams of engineers, and developing applications for adaptive training, working with scientists who encode algorithms such as Partially Observable Markov Decision Process (POMDP) and Best Fit Optimization (BFO) models. One example is a Small Business Innovative Research (SBIR) Phase II project involving developing a training platform for Full Motion Video Imagery Analysts. Ms. Kramer received a M.S. in Computer Science from Worcester Polytechnic Institute, and a B.A. in Computer Science from the University of Massachusetts, Boston. She is a member of the Association of Computing Machinery, and the national Computer Science Teachers Association.*

Chris Nucci *is a Senior Software Engineer at Aptima, Inc. with engineering experience in a variety of fields including live training, gunnery, cyber security, secure coding. His current work focuses on the ongoing development of SPOTLITE™, Aptima's tablet-based application for performance measurement, data collection, debrief, and analysis. Other work has included development on Aptima's PM Engine™ and Army's GIFT framework. His previous experience at Lockheed Martin includes development of a Storage Area Network (SAN) configuration service for the National Cyber Range (NCR), development of gunnery and gallery target and mover controller software for the Saudi Arabian National Guard (SANG) Range Modernization program, exercise planning and AAR tools for U.S. Army Program Executive Office for Simulation, Training, and Instrumentation (PEO STRI) Common Training Instrumentation Architecture (CTIA), Live Training Transformation (LT2), and Combat Training Center (CTC), as well as experience in secure coding, databases, and web application development. Mr. Nucci received a M.S. and B.S. in Computer Science from the Florida Institute of Technology.*

Proceedings of the Fifth Annual GIFT Users Symposium

GIFT, the Generalized Intelligent Framework for Tutoring, is a modular, service-oriented architecture developed to lower the skills and time needed to author effective adaptive instruction. Design goals for GIFT also include capturing best instructional practices, promoting standardization and reuse for adaptive instructional content and methods, and methods for evaluating the effectiveness of tutoring technologies. Truly adaptive systems make intelligent (optimal) decisions about tailoring instruction in real-time and make these decisions based on information about the learner and conditions in the instructional environment.



The GIFT Users Symposia began in 2013 to capture successful implementations of GIFT from the user community and to share recommendations leading to more useful capabilities for GIFT authors, researchers, and learners.

About the Editor:

Robert Sottilare, Ph.D. *is the Adaptive Training Research Lead at the US Army Research Laboratory where the focus of his research is automated authoring, instructional management, and analysis tools and methods for intelligent tutoring systems (ITSs). He is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT), an open source, AI-based adaptive instructional architecture. He is a faculty scholar and part-time professor at the University of Central Florida where he teaches a graduate level course in ITS design. Dr. Sottilare is also a frequent lecturer at the United States Military Academy (USMA) where he teaches a senior level colloquium on ITS design.*

Part of the Adaptive Tutoring Series

978-0-9977257-1-1