Proceedings of the Third Annual GIFT Users Symposium

June 2015
Orlando, Florida

Edited by:
Robert Sottilare
Anne M. Sinatra

Part of the Adaptive Tutoring Series

Proceedings of the 3rd Annual Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym3)

Conducted 17–18 June 2015 in Orlando, FL

Edited by:
Robert A. Sottilare
Anne M. Sinatra

The views expressed herein are those of the authors and do not necessarily reflect the views of the U.S. Army Research Laboratory.

Use of trade names or names of commercial sources is for information only and does not imply endorsement by the U.S. Army Research Laboratory.

This publication is intended to provide accurate information regarding the subject matter addressed herein. The information in this publication is subject to change at any time without notice. The U.S. Army Research Laboratory, nor the authors of the publication, makes any guarantees or warranties concerning the information contained herein.

Printed in the United States of America
First Printing, August 2015

U.S. Army Research Laboratory
Human Research & Engineering Directorate
SFC Paul Ray Smith Simulation & Training Technology Center
Orlando, Florida

International Standard Book Number: 978-0-9893923-8-9

We wish to acknowledge the editing and formatting contributions of Carol Johnson and Deeja Cruz

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

CONTENTS

THEME I: APPLICATIONS OF GIFT

GIFT-Powered NewtonianTalk

Weinan Zhao[1], Matthew Ventura[2], Benjamin D. Nye[3], Xiangen Hu[3,4]
[1]Florida State University, [2]Pearson Higher Education,
[3]University of Memphis, [4]Central China Normal University

INTRODUCTION

Research on learning caused by games typically focuses on games explicitly designed for learning (Tobias & Fletcher, 2011; Wilson et al., 2009). With that said, even traditional games (e.g., those not explicitly designed for learning) can also produce significant learning gains. This paper describes the implementation of an Intelligent Tutoring System (ITS) enhanced educational game called Newtonian-Talk using the Generalized Intelligent Framework for Tutoring (GIFT). This work combines elements from both ITS and educational games.

BACKGROUND

Intelligent Tutoring Systems

Meta-analyses show that ITSs produce learning effect sizes on the order of one sigma over traditional static content (Dodds & Fletcher, 2004; VanLehn, 2011), which is approximately a full letter grade in traditional grading schemes. This is halfway to the ideal goal of a two-sigma effect size (Bloom, 1984; Corbett, 2001). In the research discussed here, the AutoTutor Lite (ATL, Hu et al., 2009) ITS uses an established method of engaging a learner in a natural-language tutorial dialogue (Graesser, Chipman, Haynes, & Olney, 2005). ATL appears as an animated “talking head” avatar at certain points during the game and engages the learner in conversation about key physics concepts

We call this integration style “ITS-enhanced” because it overlays a dialogue-based ITS alongside or on-top of an existing system. This can be contrasted against an “ITS-integrated” system, where the ITS acts directly within the system with similar functionality as the learner (e.g., as an avatar in a game, for example). In an ITS-enhanced system, the tutor does not directly alter the environment, but engages with the learner to help them make better decisions. On the converse, an ITS-integrated agent could also take actions (e.g., fill in problem steps) or alter the environment to make learning activities harder or easier.

Games to Support Problem-Solving and Learning

Problem solving can be frustrating, causing some learners to abandon their problem solving practice and, hence, learning. Well-designed games can be seen as vehicles for exposing players to intellectual problem solving activities (Gee, 2004). This is where the principles of game design come in: Good games can provide an engaging and authentic environment designed to keep practice meaningful and personally relevant. With simulated visualization, authentic problem solving, and instant feedback, computer games can afford a realistic framework for experimentation and situated understanding, and thus act as rich primers for active learning (Shute & Ventura, 2013). Such support enables learners to do more advanced activities and increase their persistence for engaging in advanced thinking. The complicated part about including learning support in games is the need to balance learning with engagement and immersive flow in the gameplay. Achieving this while simultaneously reinforcing the emerging concepts and principles that deepen learning and support transfer to other contexts is non-trivial.

Physics Playground

Research into "folk" physics demonstrates that many people hold erroneous views about basic physical principles that govern the motions of objects in the world, a world in which people act and behave quite successfully (Reiner, Proffitt, & Salthouse, 2005). Recognition of the problem has led to interest in the mechanisms by which physics students make the transition from folk physics to more formal physics understanding (diSessa, 1982) and to the possibility of using video games to assist in the learning process (Masson, Bub, & Lalonde, 2011). We developed a game called Physics Playground (PP) to help middle school students understand qualitative physics (Ploetzner, & VanLehn, 1997). We define qualitative physics as a nonverbal understanding of Newton's three laws, balance, mass, conservation of momentum, kinetic energy, and gravity.

PP is a 2D sandbox game that requires the player to guide a green ball to a red balloon. In a sandbox game, the player can engage freely in activities without strict constraints imposed by the game objectives (if any exist at all). The player can nudge the ball to the left and right (if the surface is flat) but the primary way to move the ball is by drawing/creating simple machines on the screen that "come to life" once the object is drawn. Every object obeys the basic rules of physics relating to gravity and Newton's three laws of motion. Using the mouse, players draw colored objects on the screen, which become physical objects when the mouse button is released. These objects interact with the game environment according to Newtonian mechanics and can be used to move the ball. When objects interact within the game environment, they act as "agents of force" to move the ball around. The player can create simple machines such as levers, pendulums, and springboards to move the ball.

The difficulty of each puzzle is based on relative location of ball to balloon, number of obstacles present, number of agents required to solve the problem, the novelty of the problem, and other factors. More difficult problems receive higher weight as evidence to estimate the mastery level of the learner. Also, more "elegant" solutions (i.e., those using a minimal number of objects) receive greater weight to mastery level inferences. Preliminary data suggest playing PP for four hours can improve qualitative physics understanding (t (154) = 2.12, p < .05) with no content instruction or any other learning support (Shute, Ventura, & Kim, 2013).

METHODOLOGY: GIFT MANAGEMENT OF ATL AND PP

The GIFT framework provides an architecture to integrate independent learning technologies, which has been applied to act as a manager to control the behaviors of both ATL and PP, based on their real-time state information. While the vast majority of the components of an ITS may be made domain-independent, there must always be a specific component of the architecture to deal with the problems that the instructor desires to teach. Key domain-dependent components include how to assess student actions, how to implement instructional changes, and how to provide immediate feedback (Sottilare, Goldberg, Brawner, & Holden, 2012). In GIFT, these are implemented as domain-specific rules and tactics that determine how to respond to events in the learning environment (e.g., Physics Playground).

The combination of Physics Playground and AutoTutor is called NewtonianTalk. Figure 1 displays the interface of NewtonianTalk. First, an introductory explanation is provided. Below is the introductory explanation of Impulse to the player:

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

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

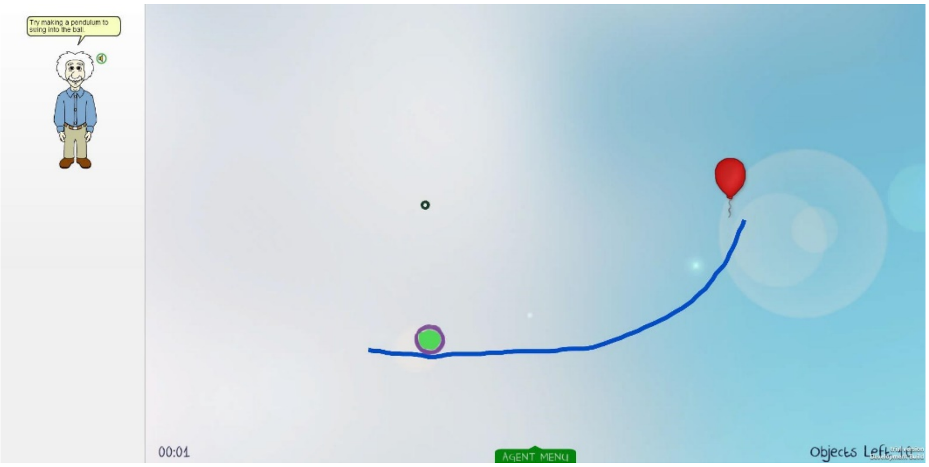


Figure 1. NewtonianTalk Interface

After the player solves all 3 puzzles in the playground, ATL poses a series of questions in natural language. Automated scores are calculated for the learner's performance. Four question and answer pairs are given for Impulse, as shown in Table 1. Once the player has answered the questions correctly or has maxed out the attempts (3 per question), the player then moves to the next playground. The player is given feedback in terms of percentages of playgrounds completed and the ATL questions.

Table 1 Questions and Answer Sets for Impulse

Question	Answer[1]
What is impulse?	*Impulse is force times time.*
How does an impulse affect an object?	*An impulse can change an object's speed.*
How could a force make a larger impulse?	*Increase the force or increase the amount of time.*
How can the same impulse be applied if the time of contact is reduced?	*To apply the same impulse over a smaller amount of time, the force must increase.*

IMPLEMENTATION

Figure 2 shows the architecture of GIFT integrated with the components of the NewtonianTalk (i.e., the integration of PP and ATL). As an overview, both PP and ATL run in the browser at the client side. They communicate with GIFT through "GIFT Web Application Infrastructure" (GIFT Web-App Infrastructure) which then communicates with the "Gateway Interoperability Plugin for Web-App" in GIFT through WebSockets. During the intervention, NewtonianTalk sends state messages from PP and ATL to GIFT. These messages include events related to performance (e.g., "elegance" of the solution in PP). GIFT assesses the state information and decides if any pedagogical strategies should be applied and sends control messages to NewtonianTalk accordingly. Assessments and pedagogical strategies, as well as conditions of applying the pedagogical strategies, are pre-configured in a Domain Knowledge File (DKF).

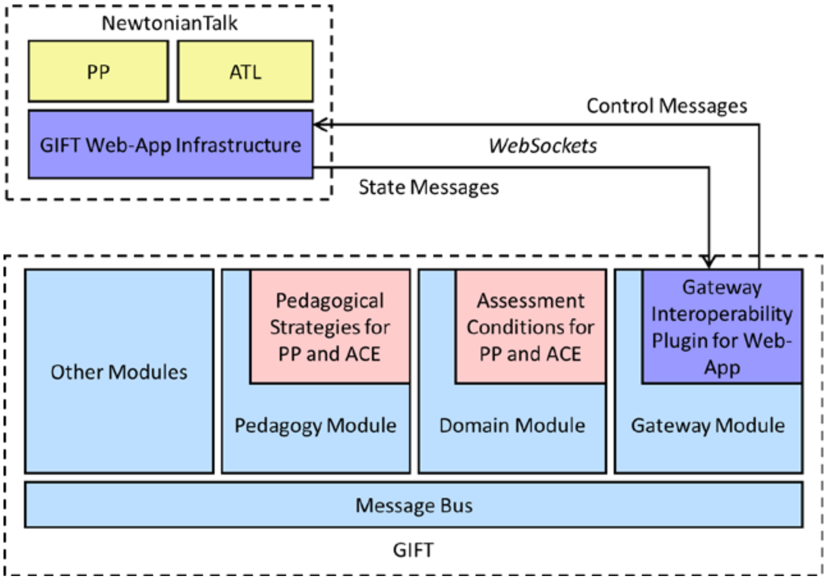

NewtonianTalk
PP
ATL
GIFT Web-App Infrastructure
Control Messages
WebSockets
State Messages
Other Modules
Pedagogical Strategies for PP and ACE
Pedagogy Module
Assessment Conditions for PP and ACE
Domain Module
Gateway Interoperability Plugin for Web-App
Gateway Module
Message Bus
GIFT

Figure 2. The Architecture of NewtonianTalk+GIFT

[1] We only use the semantic answers in the current implementation.

Communication between NewtonianTalk and GIFT

Two new GIFT components, the Interoperability Plugin and Web App Infrastructure, were added to implement the communications between NewtonianTalk and GIFT, which are shown as purple in Figure 2. Since both PP and ATL run in browsers, and full-duplex communications are needed between a browser and GIFT, WebSockets were chosen as the lower level protocol for the communications. The Interoperability Plugin starts an embedded Jetty web server with a Jetty WebSocket handler when it is enabled and waiting for a connection from the client side. On the client side, when the GIFT Web-App Infrastructure is loaded, it connects to the web server through WebSocket and begins the communication. Besides basic WebSocket communication functions, the two components also support message wrapping/unwrapping and message dispatching to different web applications, as needed. Since communication between ATL and the GIFT Web-App Infrastructure is cross domain, this component also supports cross-domain messaging in the browser (i.e., HTML5 postMessage functionality).

Assessments

In order to enable GIFT to receive state information from both PP and ATL, two message types were created for PP and ATL respectively:

1. LEVEL_STATE, which is sent to GIFT when a level in PP is ended, regardless of whether it is solved or unsolved, and sends information about how the player did in the level; and

2. SKO_STATE, which is sent to GIFT when a dialogue ends and sends information about how the player performed in the dialogue.

In addition, since some actions must be triggered when a player is at certain location in PP (e.g., upon entering a playground), a standard message type LOGIC_LOCATION is sent to GIFT each time the player's location in PP is changed. It contains the location of the player in PP, with respect to the level and state of the level. For each message, an assessment condition was created to assess the information contained in the message and return a level for the state of the player. See Appendix A for detailed information of the GIFT message types NewtonianTalk supports.

Pedagogical Strategies

Pedagogical strategies the system needs to use include starting an ATL dialogue, giving voice feedback through the talking head (but not an interactive dialogue), unlocking a playground, unlocking a level, and other interventions. A simplified but scalable method was used to implement these strategies. The GIFT "DefaultStrategyHandler" is used to deliver text feedback to the training application and a corresponding control message is included in the feedback. See Appendix B for detailed information of the control messages NewtonianTalk supports.

Currently GIFT does not explicitly support running two or more training applications concurrently and combining them in the same course. Instead, additional routing and coordination capabilities are needed. This integration is a move in that direction. This design decided against integrating PP with ATL as a single training application, before managing this single application with GIFT. This approach would require significant effort modify both apps to allow communications between them and would also limit the scalability of this scheme. Moreover, it runs against the general concept of distributed services to need to explicitly combine services first.

Instead, the logic and communications between PP and the AutoTutor Conversation Engine (ACE) are completely decoupled. Each of them only needs to interact with GIFT through the communication infrastructure described above. This design greatly increases the scalability of integrating different web apps into GIFT since each app only needs to be integrated once and can be reused in different integrations in combination with other apps. In addition, the communication components (the components colored purple in Figure 2) created for this integration can be used for web apps in general and may be of use to future web apps being integrated with GIFT.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

The design process for this integration has identified some of the technical opportunities and challenges for adding intelligent tutoring to an existing game environment that is mainly focused on simulation and experimentation. The current research integrates an ITS into a Unity game, which is a popular engine with many titles. Such games may prove powerful learning environments when intelligent tutoring is used to highlight and connect the key principles and concepts. A strength of this design has been the reliance on widely-supported web protocols to handle communication between the three systems (WebSockets and HTML5 messaging). These standards make formerly onerous or impossible communication patterns possible, including straightforward cross-domain messaging on web browsers and server push (i.e., the GIFT server sending a hint directly to the Unity instance, rather than requiring Unity to poll GIFT constantly). The Web-App communication wrappers made for this project may be useful to future researchers and training application developers seeking to integrate with GIFT.

One challenge was that GIFT itself does not offer much support for managing two learning systems explicitly. It does not allow targeting (or prioritizing) one application versus another for feedback, nor does it offer easy capabilities to make conditional or prioritization decisions based on which system a message was sent from. While this makes certain parts of integration more difficult, it is unclear if such support would be beneficial or harmful in the long term. Allowing authors to directly refer to multiple systems would make it easier to manage conditional rules and actions initially, with a lower up-front development burden. However, such rules and actions would tend to be fragile: since the name of the system would be nominal (i.e., just a name) rather than semantic (i.e., a message based on the meaning of its data), they would be unlikely to be easily portable to new system integrations. With that said, it might be useful to have metadata for messages that assign one or more types from a limited set (e.g., simulation environment, dialogue-based environment), at least for prioritization of messages with similar meanings but that have different importance.

We will be collecting data on NewtonianTalk in the Spring of 2015 on an estimated 100 undergraduate psychology students. In addition to getting valuable usability data we also will test a hypothesis regarding instruction pedagogy. The experimental manipulation will compare the instructional effectiveness of linear vs. non-linear gameplay. Giving learners more freedom to explore may enable more transferability of skills, but might also result in unproductive exploration. It is hoped that ITS support will lead non-linear conditions to be more effective overall.

ACKNOWLEDGEMENTS

This work was supported by the Army Research Lab grant W911NF-12-2-0030 and applies the open-source GIFT system as developed by the ARL Learning in Intelligent Tutoring Environments (ARL-LITE) lab and Dignitas. However, the views of this paper represent only those of the authors. This work was not sponsored by Pearson, nor does Pearson endorse the statements made in this paper. The corresponding author for this research is Xiangen Hu at the University of Memphis (xhu@memphis.edu).

REFERENCES

Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational researcher*, 4-16.

Corbett, A. (2001). Cognitive computer tutors: Solving the two-sigma problem. In User Modeling 2001 (pp. 137-147). Springer Berlin Heidelberg.

diSessa, A. A. (1982). Unlearning Aristotelian physics: A study of knowledge-based learning. *Cognitive Science*, 6, 37–75.

Dodds, P., & Fletcher, J. D. (2004). *Opportunities for new" smart" learning environments enabled by next generation Web capabilities* (No. IDA-D-2952). Institute for Defense Analyses: Alexandria, VA.

Gee, J. P. (2004). *What Video Games Have to Teach Us about Learning and Literacy*. Palgrave Macmillan.

Graesser, A. C., Chipman, P., Haynes, B. C., & Olney, A. (2005). AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. *Education, IEEE Transactions on*, *48*(4), 612-618.

Hu, X., Cai, Z., Han, L., Craig, S. D., Wang, T., & Graesser, A. C. (2009). AutoTutor Lite. In Artificial Intelligence in Education (*AIED) 2009* (pp. 802-802). IOS Press.

Masson, M. E. J., Bub, D. N., & Lalonde, C. E. (2011). Video-game training and naive reasoning about object motion. Applied Cognitive Psychology, 25, 166–173.

Ploetzner, R., & VanLehn, K. (1997). The acquisition of qualitative physics knowledge during textbook-based physics training. *Cognition and Instruction*,*15*(2), 169-205.

Reiner, C., Proffitt, D. R., & Salthouse, T. (2005). A psychometric approach to intuitive physics. *Psychonomic Bulletin and Review*, *12*, 740–745.

Shute, V. J., & Ventura, M. (2013). Measuring and supporting learning in games: Stealth assessment. Cambridge, MA: The MIT Press

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.

Sottilare, R. A., Goldberg, B. S., Brawner, K. W., & Holden, H. K. (2012, December). A modular framework to support the authoring and assessment of adaptive computer-based tutoring systems (CBTS). In *Interservice/Industry Training, Simulation, and Education Conference.(I/ITSEC) 2012*. Paper 12017 (pp. 1-13).

Tobias, S., & Fletcher, J. D. (Eds.). (2011). *Computer games and instruction*. IAP.

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

Wilson, K. A., Bedwell, W. L., Lazzara, E. H., Salas, E., Burke, C. S., Estock, J. L., & Conkey, C. (2009). Relationships between game attributes and learning outcomes review and research proposals. *Simulation & Gaming*, *40*(2), 217-266.

APPENDICES

Appendix A GIFT Message types NewtonianTalk supports

App	type	description	fields
PP	LOGIC_LOCATION	Player's location in PP. Sent to GIFT each time when a player's location in PP is changed	currentPlaygroundIndex currentLevelIndex direction
	LEVEL_STATE	Information about how a player did in a level. Sent to GIFT when a level in PP is ended, no matter it is solved or unsolved.	domainSessionId playgroundIndex levelIndex solved solutionData
ATL	SKO_STATE	Information about how a player did in an SKO. Sent to GIFT when an SKO is ended	domainSessionId nodeId finished data

Appendix B Control message types NewtonianTalk supports

App	type	Description	Fields	
			object	params
PP	PAUSE_PP	Pause the game	PHYSICSPLAYGROUND	delay
	RESUME_ PP	Resume the game	PHYSICSPLAYGROUND	delay
	UNLOCK_ LOCATION	Unlock locations in PP.	PHYSICSPLAYGROUND	locations: e.g.: 0.1;0.2;1;1.* 1: unlock playground 1 1.*: unlock all the puzzles in playground 1
	SET_ PUZZLE_ SCORE	Set the score of a puzzle	PHYSICSPLAYGROUND	location solved score (0-1)
ATL	LOAD_SKO	Start an SKO	AUTOTUTOR	guide nodeId
	SET_SKO_ SCORE	Set the score of an SKO	AUTOTUTOR	location type solved score
	STOP_ CURRENT_SKO	Stop current SKO	AUTOTUTOR	
	RESUME_ CURRENT_SKO	Resume current SKO	AUTOTUTOR	
	SKO_TALK	Have the talking head to say some text. If “type” is alternative, the text to be said will be randomly selected from the array.	AUTOTUTOR	type: simple/alternatives content: simple: string alternatives: array. e.g., ["", "", ""]
	SHOW_SKO_ CONTROLS	Show the SKO control buttons (stop button)	AUTOTUTOR	
	SHOW_ SKO_ BUTTON	Show an SKO start button.	AUTOTUTOR	text guide nodeId canClose:true/false
	REMOVE_ SKO_ BUTTON	Remove the SKO start button.		

ABOUT THE AUTHORS

Mr. Weinan Zhao *is a PhD candidate of Instructional Systems and Learning Technology at the Florida State University. His research interests are game-based learning and assessment, knowledge modeling, educational data mining and intelligent tutoring systems. Weinan is the lead developer of NewtonianTalk.*

Dr. Matthew Ventura *is a Senior Learning Designer at Pearson Education. Before coming to Pearson, Matthew was a Senior Research Scientist at Florida State University where he co-lead the design of Physics Playground.*

Dr. Benjamin Nye *is a research assistant professor at the University of Memphis Institute for Intelligent Systems (IIS). His research interests include modular intelligent tutoring system designs, modeling social learning and memes, cognitive agents, and educational tools for the developing world. He received his Ph.D. in Systems Engineering from the University of Pennsylvania in 2011. Ben led work on the Sharable Knowledge Objects (SKO) framework, a service-oriented architecture for AutoTutor. He is also researching and data mining a large corpus of human-to-human online tutoring dialogues, as part of the ADL Personalized Assistant for Learning (PAL) project. Ben's major research interest is to identify barriers and solutions to development and adoption of ITS so that they can reach larger numbers of learners, which has traditionally been a major roadblock for these highly-effective interventions.*

Dr. Xiangen Hu *is a professor of Psychology and Electronic and Computer Engineering, senior researcher in the Institute for Intelligent Systems (IIS) at the University of Memphis. Dr. Hu currently is a visiting professor of Psychology at the Central China Normal University. He is the director of the Advanced Distributed Learning (ADL) Center for Intelligent Tutoring Systems (ITS) Research & Development (ADL-C-ITS-R&D) and leads the team at ADL-C-ITS-R&D that integrates the AutoTutor Framework with GIFT.*

Using GIFT to Model and Support Students' Metacognition in the UrbanSim Open-Ended Learning Environment

James R. Segedy[1], John S. Kinnebrew[1], Benjamin S. Goldberg[2], Robert A. Sottilare[2], Gautam Biswas[1]
[1]Institute for Software Integrated Systems, Department of Electrical Engineering and Computer Science
[2]U.S. Army Research Laboratory – Human Research and Engineering Directorate, Simulation and Training Technology Center (STTC)

INTRODUCTION

Open-ended computer-based learning environments (OELEs) (Land, Hannafin, & Oliver, 2012; Segedy, Kinnebrew, & Biswas, in press) are learner-centered environments that present students with a challenging problem-solving task, information resources, and tools for completing the task. Students are expected to use the resources and tools to make decisions about how to proceed and in this process learn about the problem domain while developing strategies for task completion and problem-solving. In OELEs, students have to distribute their time and effort between exploring and organizing their knowledge of the problem domain, creating and testing hypotheses, and using their learned knowledge to make progress toward goals (which are often problems that have multiple solution paths). Since there are no prescribed solution steps, students may have to discover the solution process using exploratory methods. Moreover, there may not be an obvious, single "best" solution to a problem. Therefore, the exploration process may require students to consider trade-offs and employ their critical thinking and evaluation skills that progressively lead them to achieving a good solution.

Succeeding in OELEs can be difficult because of the cognitive and metacognitive demands that such environments place on learners. To solve problems as they learn about a new domain, students have to simultaneously wrestle with their emerging understanding of a complex topic, develop and utilize skills to enable and support their learning, and employ self-regulated learning (SRL) (Zimmerman & Schunk, 2011) processes for managing the open-ended nature of the task. As such, OELEs can prepare students for future learning (Bransford & Schwartz, 1999) by developing their ability to independently investigate and develop solutions for complex open-ended problems.

In this paper, we discuss our recent work in integrating the UrbanSim counter-insurgency (COIN) command simulation (McAlinden, Durlach, Lane, Gordon, & Hart, 2008) and the Generalized Intelligent Framework for Tutoring (GIFT). The goal of this work is to develop general representations and authoring tools for monitoring and supporting students' *metacognitive thinking*, a vital component of SRL, while using OELEs such as UrbanSim. Metacognition (Brown, 1975; Flavell, 1976) describes the ability to reason about and explicitly manage one's own cognitive processes. In particular, our work is centered on students' understanding and use of *strategies*, which have been defined as consciously-controllable processes for interpreting, analyzing, and completing tasks (Pressley, Goodchild, Fleet, Zajchowski, & Evans, 1989).

Towards this end, we have recently conducted a study with students from a University Reserve Officers' Training Corp (ROTC) program who used the UrbanSim OELE through GIFT as part of their regular classroom activities. Before and after students used the system, we asked them to explain their understanding of COIN and the steps involved in conducting a COIN operation. We present a qualitative analysis of our initial results from this study that shows students learned about COIN doctrine and also developed or refined their strategies for information acquisition and planning to support COIN operations in a realistic scenario (in the UrbanSim simulation environment). These early results are encouraging, and we expect to gain valuable insight from the collected data, such as how students' strategies evolved as they worked with UrbanSim over the course of the study. Our overall goal is to develop a metacognitive tutor for UrbanSim using the GIFT authoring tools that will support and help students learn metacognitive problem solving strategies when working on complex COIN problems in the UrbanSim environment.

BACKGROUND

Metacognition (Flavell, 1976) describes the ability to reason about and explicitly manage one's own cognitive processes. It is often broken down into two sub-components: knowledge and regulation (Schraw, Crippen, & Hartley, 2006; Young & Fry, 2008). Metacognitive knowledge refers to an individual's understanding of their own cognition and strategies for managing that cognition. Metacognitive regulation refers to how metacognitive knowledge is used for creating plans, monitoring and managing the effectiveness of those plans, and then reflecting on the outcome of plan execution in order to develop and refine metacognitive knowledge (Veenman, 2011).

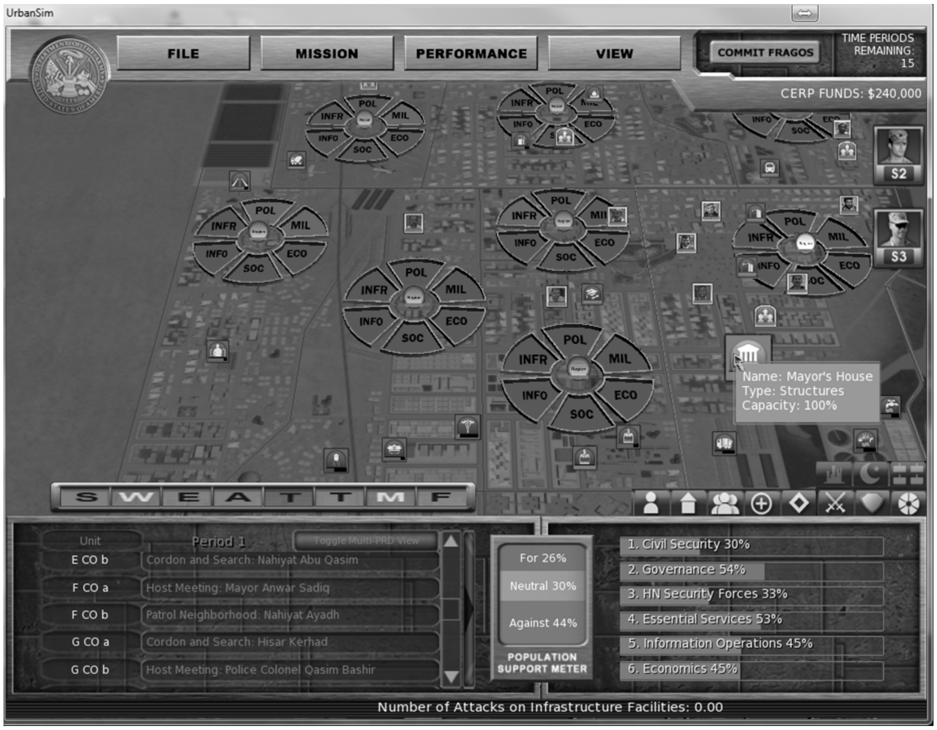
When applied to learning, metacognition can be considered a subset of self-regulated learning (SRL). SRL is a theory of active learning that describes how learners are able to set goals, create plans for achieving those goals, continually monitor their progress, and revise their plans to make better progress in achieving the goals (Zimmerman & Schunk, 2011). In terms of SRL, metacognition deals directly with cognition without explicitly considering its interactions with emotional or motivational constructs (Whitebread & Cárdenas, 2012). Despite this separation, models of self-regulation are valuable in depicting key metacognitive processes. For example, Roscoe, Segedy, Sulcer, Jeong, and Biswas (2013) describe SRL as containing "multiple and recursive stages incorporating cognitive and metacognitive strategies" (p. 286). This description of SRL involves phases of orientation and planning, enactment and learning, and reflection and self-assessment.

Our focus on metacognition is centered on students' understanding and use of *strategies*, which have been defined as consciously-controllable processes for completing tasks (Pressley et al., 1989). Strategies comprise a large portion of metacognitive knowledge; they consist of declarative, procedural, and conditional knowledge that describe the strategy, its purpose, and how and when to employ it (Schraw et al., 2006). The research community has identified several types of strategies based on the tasks for which they are designed. Strategies may be cognitive (e.g., a strategy for applying a particular procedure, or completing an addition problem), metacognitive (e.g., strategies for choosing and monitoring one's own cognitive operations), involve management (e.g., for managing one's environment to promote focused attention), be directed toward learning (e.g., a strategy for memorizing new information), or involve a combination of these (Pressley et al., 1989). For example, a metacognitive learning strategy might involve *activating prior knowledge* before reading about a topic by consciously bringing to mind information one already knows about the topic (Bouchet, Harley, Trevors, & Azevedo, 2013). When faced with a complex task, students must either identify/adapt a known strategy for completing it or invent one using their metacognitive knowledge.

An important characteristic of a strategy is its *level of generality*. That is, some strategies apply to very specific situations (e.g., an approach to adding two-digit numbers) while other strategies apply to a broader set of situations (e.g., summarizing recently learned information to improve retention). An understanding of more general strategies, as well as their specific implementations for concrete tasks, is important for developing one's ability to adapt existing strategies to new situations or invent new strategies. Thus, our goal in GIFT is to explicitly teach students general strategies and help students understand how to apply them to complex tasks. In the longer run, as students encounter different situations in which a strategy applies, we hope to make students aware of how strategies, such as those for maintaining situational awareness, monitoring the execution of one's plan, and evaluating the benefits and drawbacks of a previously-executed task, may generalize across tasks and domains (Bransford & Schwartz, 1999). A pre-requisite to achieving this goal in the GIFT framework is the ability to conceptualize and build domain-independent structures for representing metacognitive strategies and processes.

THE URBANSIM OPEN-ENDED LEARNING ENVIRONMENT

UrbanSim (McAlinden et al., 2008), shown in Figure 1, is a turn-based simulation environment in which users assume command of a COIN operation in a fictional Middle-Eastern country. Users have access to a wealth of information about the area of operation, including: intelligence reports on key individuals, groups, and structures; information about the stability of each district and region; economic, military, and political ties between local groups in the region; the commanding team's current level of population support; and the team's progress in achieving six primary lines of effort. Users have a limited amount of resources at their command to perform the COIN operations, and the actions that users take are scenario-specific. These actions generally have to be directed toward increasing the area's stability by making progress along the different lines of effort: (1) improving civil security; (2) improving governance; (3) improving economic stability; (4) strengthening the host nation's security forces; (5) developing and protecting essential services and infrastructure; and (6) gaining the trust and cooperation of the population.



TIME PERIODS REMAINING: 15

CERP FUNDS: $240,000

Name: Mayor's House
Type: Structures
Capacity: 100%

Unit	Period 1
E CO b	Cordon and Search: Nahiyat Abu Qasim
F CO a	Host Meeting: Mayor Anwar Sadiq
F CO b	Patrol Neighborhood: Nahiyat Ayadh
G CO a	Cordon and Search: Hisar Kerhad
G CO b	Host Meeting: Police Colonel Qasim Bashir

POPULATION SUPPORT METER
For 26%
Neutral 30%
Against 44%

1. Civil Security 30%
2. Governance 54%
3. HN Security Forces 33%
4. Essential Services 53%
5. Information Operations 45%
6. Economics 45%

Number of Attacks on Infrastructure Facilities: 0.00

Figure 1. UrbanSim

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

UrbanSim provides documentation and tutorials that help students gain an appreciation for the challenges inherent in managing COIN operations. For example, they should learn the importance of maintaining situational awareness, managing trade-offs, and anticipating 2nd- and 3rd-order effects of their actions, especially as the game evolves (McAlinden et al., 2008). They should also understand that their actions themselves produce intelligence and, therefore, they need to continually "*learn and adapt*" in such complex domains with overwhelming, yet incomplete, information. In other words, students should realize that their decisions result in new information that may be critical for decision making and planning during upcoming turns. Students can learn about the effects of their actions by viewing causal graphs provided by their intelligence and security officer (S2). Users who adopt strategies to better understand the area of operation and its culture by viewing and interpreting the effects of their actions using these causal graphs should progressively make better decisions in the simulation environment as the COIN scenario evolves.

UrbanSim Learner Model

To represent metacognition in GIFT, we are currently designing extensions to its learner modeling capabilities. In GIFT, a learner model consists of a set of named *concepts* that are assessed continually while students are interacting with designated course materials. At any time, each concept may be assessed as below, at, or above expectation, and higher-level concepts may be related hierarchically to lower-level concepts. Thus, a *basic mathematics* concept may be based on assessments of its component concepts: *addition*, *subtraction*, *multiplication*, and *division*. The data representation is similar to the sampling of a stream: GIFT monitors each student's task performance over time, updating the concept assessments based on his or her most recent performance. Thus, a student may perform above expectation on one subtraction problem and below expectation on the next. A history of these assessments is maintained for feedback purposes, both during and after learning.

Building from this, we will employ a three-level framework for analyzing and representing students' skill and strategy proficiency, shown in Figure 2. Analyses will involve making a set of inferences based on students' observed behaviors while using the learning environment. In this framework, direct observations of students' behaviors will serve as assessments of their ability to correctly execute strategies. To accomplish this, we are designing tools that allow course designers to specify how actions can be combined to enact a strategy. The resulting strategy model can be used online to interpret students' action sequences in terms of these strategies. The strategy that best matches students' behaviors will be assumed to be the strategy they are applying, and further assessments will examine whether or not they execute the strategy correctly. For example, a student may be using a monitoring strategy in which they check their recently-completed work to make sure it is correct. However, they may erroneously conclude that their work was completed correctly, indicating an ineffective execution of the strategy.

These assessments and interactions, along with continued monitoring of student behavior, will serve as the basis for assessing students' understanding of domain-specific strategies. When GIFT observes a correctly-executed strategy, it will increase its confidence in the student's understanding of the associated procedural knowledge. Similarly, when GIFT observes a strategy executed at an appropriate time, it will increase its confidence in the student's understanding of the associated conditional knowledge. To test students' declarative knowledge, GIFT will interact with them directly through conversational assessment techniques to better infer their understanding.

The final (top) level of our framework involves linking students' understanding of task-specific strategies to task-general representations of those strategies. An important aspect of metacognition is that the declarative knowledge for a number of strategies can be expressed in a domain-general form. In other words, many strategies can be applied to multiple tasks, situations, and contexts. Our approach leverages this property by developing new GIFT capabilities for tracking and supporting task-general strategies in multiple contexts. When a student correctly employs a task-specific strategy, GIFT will link this use of the strategy to its task-general representation and the context in which the strategy was applied. This information will be stored in GIFT's long-term learner model and can be referenced during future learning sessions. The goal is to integrate information about strategy use across multiple contexts, allowing GIFT to provide instruction and guidance that draws connections between a learner's current tasks and their previous experiences. For example, GIFT could guide the student through an analogy:

"*This task is just like when you had to do [X] in [ENV] back in [MONTH]. The main difference is [Y].*" Next steps will involve defining pedagogical representations that are informed by this learner modeling approach, and how guidance functions will be managed across the varying levels of abstraction.

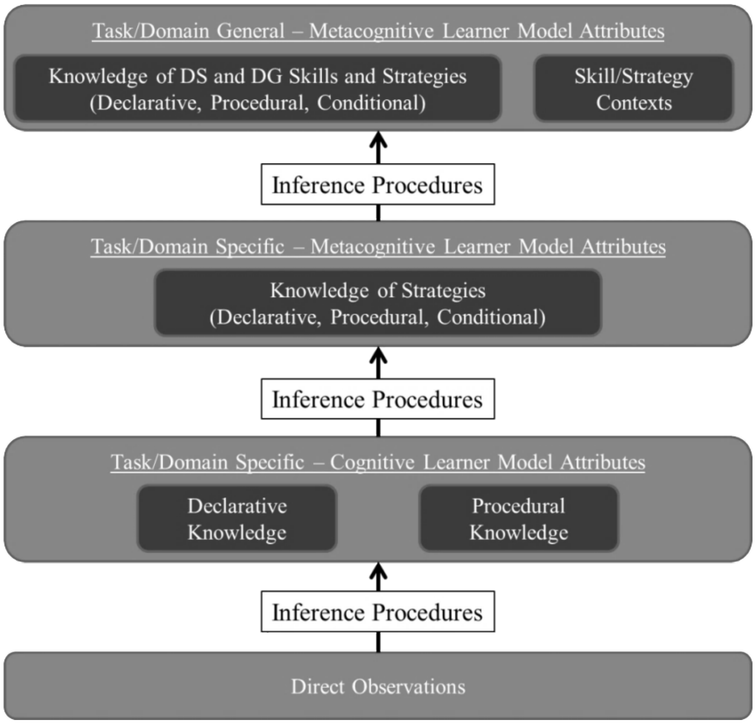

Task/Domain General – Metacognitive Learner Model Attributes
Knowledge of DS and DG Skills and Strategies (Declarative, Procedural, Conditional)
Skill/Strategy Contexts
Inference Procedures
Task/Domain Specific – Metacognitive Learner Model Attributes
Knowledge of Strategies (Declarative, Procedural, Conditional)
Inference Procedures
Task/Domain Specific – Cognitive Learner Model Attributes
Declarative Knowledge
Procedural Knowledge
Inference Procedures
Direct Observations

Figure 2. Skill and Strategy Detection Framework

Connecting UrbanSim to GIFT

In order to connect UrbanSim to GIFT, we created a Java application that monitors the UrbanSim log files and publishes the information to any interested parties. The various components and their interactions necessary for connecting UrbanSim and GIFT are shown in Figure 3. UrbanSim produces log files that include information on the actions taken in the simulation and the effects of those actions. Our log parser reads these files as they are created and communicates the *actions* taken by learners as well as the *contexts* in which those actions occur. In this instance, a context can be considered to be equivalent to an interface configuration. For example, the configuration shown in Figure 1 displays a map of the area of operation. By tracking the actions and contexts logged by UrbanSim, we are able to create a detailed understanding of students' behaviors in the program.

In the tutor we used during our preliminary study, we configured GIFT to detect when students commit their orders and present them with a survey through GIFT's tutor user interface, as shown in Figure 4. The survey asked students the following questions:

- What were your goals when you committed these FRAGOs?
- How did you expect these FRAGOs to help you achieve your goals?

- What trade-offs or negative effects did you expect as a result of these FRAGOs?
- How was your approach this turn different from your last turn (if applicable)?
- Did your FRAGOs have the effect you had intended? Why or why not?
- Were the outcomes of your FRAGOs the same, worse, or better than what you expected? Why do you think that?
- If offered another opportunity, what would you do differently on the turn you just completed?

We expect that the data collected through this survey will provide valuable insights into how students analyze situations in UrbanSim and learn from them as the simulation progresses.

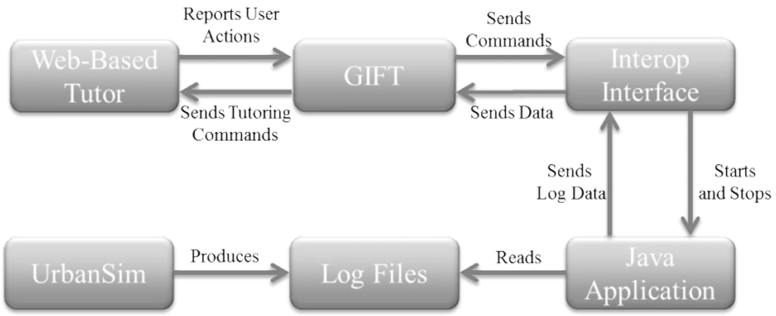


Figure 3. Communication between GIFT and UrbanSim

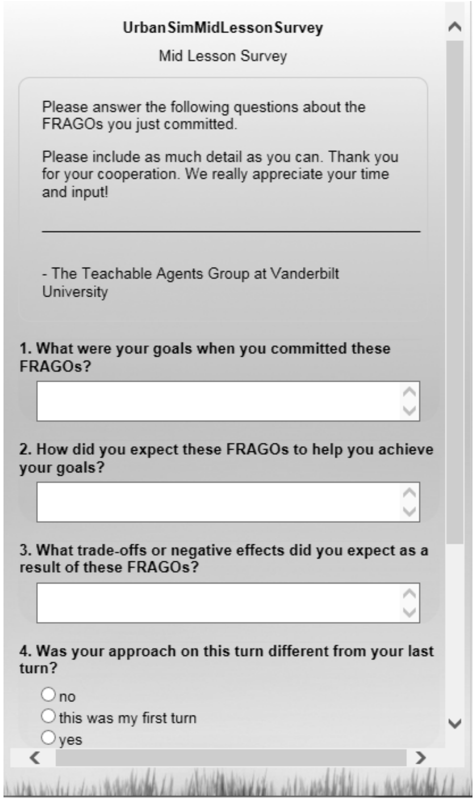
PRELIMINARY STUDY OF ROTC STUDENTS USING URBANSIM

The goal of our preliminary study of ROTC students was to collect data on students' use of the system. In particular, we were interested in data about students' goals, approaches, strategies, expectations, surprises, interpretations, and understanding of COIN concepts. This will help us identify the primary metacognitive strategies that we will incorporate into a metacognitive GIFT tutor for UrbanSim.

Fourteen senior-year ROTC students from a Southeastern United States university participated in the study. These students worked in pairs during two separate 2-hour sessions (approximately one month apart). Due to absences, these 14 students comprised 8 groups during the two sessions, with 4 groups remaining the same for both sessions. Students used UrbanSim to practice COIN in two scenarios: Al-Hamra and Al-Hamra 2. In both scenarios, they had access to intelligence on the area of operation, including its key structures, individuals, and groups. Their mission in the Al-Hamra scenario was to "restore Al-Hamra's civil infrastructure and ensure the happiness and productiveness of the population." In Al-Hamra 2, they were charged with establishing "successful government and security that meets the needs of its citizens while stopping the flow of extra-national insurgents."

The study proceeded as follows: (1) students completed a pre-activity survey asking them about their experience with and understanding of COIN; (2) students used UrbanSim with the Al-Hamra 2 scenario for approximately 90 minutes; (3) students used UrbanSim with the Al-Hamra scenario for approximately

90 minutes; (4) the course professor led a debriefing discussion with the students; and (5) students completed a post-activity survey. The surveys included two questions analyzed in this paper: (1) Please briefly explain, in your own words, how the Clear-Hold-Build doctrine applies to counter-insurgency operations; (2) Imagine that you have been assigned command of a counter-insurgency operation. Explain the steps you would go through before formulating your own mission plan and lines of effort.



UrbanSimMidLessonSurvey

Mid Lesson Survey

Please answer the following questions about the FRAGOs you just committed.

Please include as much detail as you can. Thank you for your cooperation. We really appreciate your time and input!

- The Teachable Agents Group at Vanderbilt University

1. What were your goals when you committed these FRAGOs?

2. How did you expect these FRAGOs to help you achieve your goals?

3. What trade-offs or negative effects did you expect as a result of these FRAGOs?

4. Was your approach on this turn different from your last turn?

○ no
○ this was my first turn
○ yes

Figure 4. UrbanSim survey presented through GIFT

During the study, we collected several streams of data, including: (1) log files from UrbanSim and GIFT; (2) pre- and post- activity surveys and post-turn surveys; and (3) audio-video data from the computer web-cams synced to a screen capture video. In the next section, we present a preliminary qualitative analysis of the pre- and post- activity survey responses.

PRELIMINARY RESULTS

In analyzing students' survey data, our primary finding was that after students used GIFT and UrbanSim, their survey answers, in general, became more specific. Thus, there is evidence that the experience with GIFT and UrbanSim helped students gain a more concrete understanding of the essence of COIN operations and how they evolve over time. Further, the additional details in the COIN operations planning question on the post-survey suggest that students developed or refined strategies for COIN information acquisition and planning. To illustrate these preliminary findings, we present selected answers from the two survey questions listed in the previous section.

Explaining Clear-Hold-Build

Clear-Hold-Build (CHB) is a counter-insurgency strategy with three distinct phases. First, military forces *clear* an area of insurgents. Second, they focus on *holding* the cleared area and preventing further insurgent infiltration. Third, they focus on *building* up the area's government, police forces, and infrastructure such that the local population is able to safeguard the area independently.

Most students demonstrated a basic understanding of these principles during both the pre-survey and post-survey. However, four students indicated that they did not know what clear-hold-build was during the pre-survey. Typical answers to this survey question mentioned the three phases and included a high-level description of each phase. Pre-survey answers included the following:

- User A: Clear: taking over control of a certain area – eliminating enemy presence. Hold: maintain a presence in the community. Build: restore important economic infrastructure & build up the civilians' trust in our mission.

- User B: Clear-Hold-Build is a progressive method for establishing security and stability over a given area by first conducting patrols, cordons and other military operations to secure the area, maintain a constant presence and gradually construct vital institutions – governance, infrastructure, police, public works, etc. – to gain support of the population.

All students were able to explain CHB during the post-survey with about the same level of detail as on the pre-survey:

- User A: Clear - eliminate the enemy; Hold - maintain presence in area; Build - reconstructive efforts.

- User B: An area must initially be cleared of insurgents in a dynamic military operation, then security must be established and control imposed within an area. Finally infrastructure and essential services need to be built within the area to secure the support of the population.

Overall, students who were unfamiliar with CHB on the pre-survey understood it after using GIFT and UrbanSim.

Explaining Steps for Formulating a Counter-Insurgency Plan

During the pre-survey, students' answers to this question were fairly general; they could apply to almost any mission and often lacked focus and prioritization relevant to the task. Typical answers included:

- User C: Understand civilian support of insurgents and coalition forces. Understand the general conditions of the area & its importance.

- User D: I would discuss w/my command team our intent and the best way to complete our mission.

One student did provide a more involved pre-survey answer: (1) Where am I? I need to know the history, geography, and cultural norms of the locals. Who are the local leaders? (2) What do the insurgents want? What do they use as recruitment tools? (3) What forces, both US & local, are available to me? Do I have adequate funding?

Compared to the pre-survey, answers on the post-survey were more detailed. These answers included:

- User C: I would investigate how strong the insurgents are and where they seem to be centralized and then meet w/local leaders to see what lines of effort are most important to them.

- User D: I would hold meetings w/key leaders in area, as well as soldiers/staff that are familiar w/area and then decide on areas and priorities to focus on and then start my plan.

These answers are more specific than those from the pre-survey, indicating that students may have developed more concrete, focused strategies for planning and information acquisition in COIN operations.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

In this paper, we have described our approach for developing and incorporating a metacognitive tutoring framework into the GIFT platform. Our approach leverages research on metacognition, strategies, and strategy instruction with open-ended learning environments to track students' understanding and use of strategies for complex problem-solving scenarios. Moving forward, we will continue developing this framework, incorporate it into GIFT, and instantiate it with the UrbanSim counter-insurgency simulation.

The results of our preliminary data analysis indicate that students did learn as a result of using GIFT and UrbanSim, including developing and refining strategies for COIN operations. This early positive indication is encouraging, and we expect to gain valuable insights into students' strategy uses from the rich data we have collected in the study. We plan on identifying strategies used by students and working with ROTC instructors to identify which of these strategies are more and less effective in general, and why. We are currently investigating the following: (1) How well do students utilize the information available to them in UrbanSim? (2) What strategies do students understand and execute effectively and ineffectively? (3) What do student pairs discuss as they use the simulation, and how does this provide indications of their understanding of strategies? (4) How well do their Lines of Effort (LOE) priorities align with the goals of the mission, and which LOEs do students focus on and ignore?

In parallel, we are working with GIFT developers to begin implementing our metacognitive framework for strategy identification and ultimately use students' current behavior and prior experiences to direct scaffolding. Our analysis of protocols of student discussions as they worked on UrbanSim will provide us with initial data on use of strategies and the context in which they are applied. More advanced analyses will involve the application of analytic measures and sequence mining techniques to develop and refine strategy identification methods over time, which in turn will better inform state representations that trigger scaffolding. These combined analyses will allow for powerful tutoring interactions between students and GIFT.

REFERENCES

Bouchet, F., Harley, J., Trevors, G., & Azevedo, R. (2013). Clustering and profiling students according to their interactions with an intelligent tutoring system fostering self-regulated learning. *Journal of Educational Data Mining, 5*(1), 104-146.

Bransford, J., & Schwartz, D. (1999). Rethinking transfer: A simple proposal with multiple implications. *Review of Research in Education, 24*(1), 61-101.

Brown, A. (1975). The development of memory: Knowing about knowing, and knowing how to know. In H. Reese (Ed.), *Advances in Child Development and Behavior* (pp. 103-152), New York, NY: Academic Press.

Flavell, J. (1976). Metacognitive aspects of problem solving. In L. Resnick (Ed.), *The Nature of Intelligence* (pp. 231-236). Hillsdale, NJ: Erlbaum.

Land, S., Hannafin, M., & Oliver, K. (2012). Student-centered learning environments: Foundations, assumptions and design. In D. Jonassen & S. Land (Eds.), *Theoretical Foundations of Learning Environments* (pp. 3-25). New York, NY: Routledge.

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.

Pressley, M., Goodchild, F., Fleet, J., Zajchowski, R., & Evansi, E. (1989). The challenges of classroom strategy instruction. *The Elementary School Journal, 89*, 301-342.

Roscoe, R.D., Segedy, J.R., Sulcer, B., Jeong, H., & Biswas, G. (2013). Shallow strategy development in a teachable agent environment designed to support self-regulated learning. *Computers & Education, 62*, 286-297.

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

Segedy, J.R., Kinnebrew, J.S., & Biswas, G. (in press). Using coherence analysis to characterize self-regulated learning behaviours in open-ended learning environments. *Journal of Learning Analytics*.

Veenman, M. (2011). Learning to self-monitor and self-regulate. In R. Mayer & P. Alexander (Eds.), *Handbook of Research on Learning and Instruction* (pp. 197-218). New York, NY: Routledge.

Whitebread, D., & Cárdenas, V. (2012). Self-regulated learning and conceptual development in young children: The development of biological understanding. In A. Zohar & Y.J. Dori (Eds.), Contemporary Trends and Issues in Science Education: Vol. 40. *Metacognition in Science Education: Trends in Current Research* (pp. 101-132). Netherlands: Springer Science+Business Media.

Young, A. & Fry, J. (2008). Metacognitive awareness and academic achievement in college students. *Journal of the Scholarship of Teaching and Learning, 8*(2), 1-10.

Zimmerman, B., & Schunk, D. (Eds.). (2011). *Handbook of Self-Regulation of Learning and Performance*. New York, NY: Routledge.

ABOUT THE AUTHORS

***Dr. James R. Segedy** is a Postdoctoral Research Scholar in the Teachable Agents Group at Vanderbilt University. His research focuses on developing measures of open-ended and independent problem solving behaviors for use by classroom teachers. His work has also included developing adaptive scaffolds such as contextualized conversational feedback and guided practice in open-ended learning environments.*

***Dr. John S. Kinnebrew** is a Research Scientist at the Institute for Software Integrated Systems at Vanderbilt University. He is currently involved in a variety of computer-based learning environment projects, where his research focuses on data mining and machine learning for modeling human learning behaviors, including metacognition and self-regulated learning strategies.*

***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 US Army Research Laboratory's Learning in Intelligent Tutoring Environments (LITE) Lab in Orlando Florida. He is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT).*

***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. He conducts research in Intelligent Systems with primary interests in simulation, and analysis of complex embedded systems, their applications to diagnosis, prognosis, and fault-adaptive control, and the design and development of Computer-Based Learning Environments. His research has been supported by funding from ARL, NASA, NSF, DARPA, and the US Department of Education.*

Theme II: Learner Modeling

Developing Persistent, Interoperable Learner Models in GIFT

Gregory A. Goodwin[1], Jennifer S. Murphy[2], Michael Hruska[3]
U.S. Army Research Laboratory – Human Research and Effectiveness Directorate[1],
Quantum Improvements Consulting[2], Problem Solutions[3]

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.

Assessments in the Learner Model

ITSs can adapt training for both an inner (micro-adaptation) and outer (macro-adaptation) loop (VanLehn, 2011). Assessments in the learner model support both of these kinds of adaptation. The inner loop pertains to the tutor-student interaction during training and depends heavily on measures of learner performance taken at that time. The outer loop supports macro-adaptivity. For example, the outer loop might choose what block of instruction is trained or perhaps the instructional strategy. This macro-adaptation depends more heavily on measures of student proficiencies, aptitudes, and goals. Table 1 summarizes the kinds of measures used for micro- and macro-adaptation.

As can be seen in Table 1, the assessments that affect macro-adaptation are trait-like and can be made before the training. One way of thinking of this is that these are the factors that the learner brings to the training. These include content-dependent factors like prior experience and training as well as things that are content-independent like intellect and personality. The measures that affect micro-adaptation are done at run-time. These are measures of what the training does to the learner. They also have a content dependent category like assessments of comprehension or skill improvement and content-independent measures like boredom or fatigue.

It would be expected that the trait-like measures would have an impact on the state-like measures. 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).

Persistent Learner Models

It is important to think about how a learner model is created and sustained. The state-like measures must be taken at run-time, but the trait-like measures could be obtained from other systems (e.g., learning-, training-, and personnel-management). Alternatively, the ITS could administer a battery of measures of experience, personality, and aptitude, but this would be very time-consuming and would detract from the advantages of being able to adapt the training based on those factors.

In fact, 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.

Table 1. Components of the Learner Model.

	Learner Measure Category	Trait-Like (macro-adaptation)	State-Like (micro-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

Both of these circumstances indicate that the learner model needs to exist and be updated independent of a single adaptive training system. This model of the learner must encompass skill/expertise development across the entire course and have measures of learner performance from most if not all of the periods of instruction.

In this paper, we propose a framework for the development and use of persistent learner models that could operate across training platforms. We recommend developing a capability within the Generalized Intelligent Framework for Tutoring (GIFT) to consume performance measures stored using the Experience API (xAPI) standard and then to utilize those xAPI statements to develop an interoperable and persistent learner model. Eventually other adaptive training systems should be able to do the same and collectively refine the common learner model.

Learner Competencies

Competence is the ability to do a job well. Competency is the set of knowledge, skills, and abilities that comprise competence in a specific job or role. Organizations define competencies in different ways. For example, the Army identifies twelve 21st century competencies and attributes. It further breaks these down into various learning outcomes, course outcomes, and learning objectives (U.S. Army Training and Doctrine Command 2011). For the purposes of this paper, we define a competency as specialized, job-specific skills, knowledge, and abilities that are developed over time and can be a result of both institutional and on-the-job training.

Educational researchers have developed frameworks which are capable of describing competencies. These frameworks are content (or in GIFT, domain[2]) independent. Such models were originally conceived by Bloom, Gagné, Posner, and others. These frameworks could be used to map competencies into an interoperable learner model.

Learning Domains

Bloom (1956) developed a well-known framework for describing competencies that includes three learning domains: cognitive, psychomotor, and affective. Bloom subdivided the cognitive domain into three categories: factual, problem solving, and procedural. Others have updated and revised Bloom's categories, most significantly adding metacognitive to this list (Anderson, et al.,, 2001). Others, including Gagné (1989) and Clark (Clark & Chopeta, 2004) have suggested alternative sets of cognitive sub-domains. Still others would argue that there is a fourth, social domain (e.g., Soller, 2001) though we will limit our discussion to the first three in this paper.

[2] The term "domain" is applied differently by GIFT than by Bloom. In GIFT, this term refers to the content being delivered. When Bloom and others refer to domains of learning, they are referring to broad areas of affective, cognitive, and psychomotor learning.

Levels of Performance

Development of expertise/skill/ability within each of these domains is seen as progressing through several levels. For complex domains or skills, progressing to the highest levels of performance may take years of training or practice. In many ways, these levels can be thought of as stages of development. Most attention has been focused on the cognitive domain. The levels/stages are generally described in terms of learning outcomes.

For example, in the cognitive domain, the most recognizable model is Bloom's 6 stage taxonomy (knowledge, understanding, application, analysis, synthesis, and evaluation). These levels were later revised to: remembering, understanding, applying, analyzing, evaluating, and creating (Anderson et al., 2001).

Levels within the Psychomotor domain have been described by Simpson (1972) and Dave (1970). Fitts and Posner (1967) propose a three stage model of expertise development. The first stage is the Declarative Stage (understanding of the task). The second stage is the Associative stage (conscious effort to execute the skill correctly). The third stage is the Automaticity stage (skill is executed with little conscious effort).

Finally, the affective domain has to do with attitudes, beliefs, values, emotions, opinions, and motivation. Perhaps the best known description of the affective domain comes from Krathwohl (Krathwohl, Bloom & Masia, 1973) who proposed the following five levels of the affective domain: Receiving (awareness, willingness to hear), Responds (react, comply), Valuing (appreciates, behavior often impacted by values), Organization (prioritize values, resolve conflicts between values), and finally Internalized (consistent and pervasive impact of values).

The many different competing ways of describing different competency domains or sub-domains as well as "levels" within each competency domain, make it difficult to establish a universal, standard framework for use with training systems. Although these competing frameworks are largely theoretical constructions, the emergence of a standard framework should be based on evidence. That is, there should be as many domains and levels as are necessary to effectively adapt training.

Referring back to the components of the learner model described in Table 1, it can be seen that the three domains (cognitive, psychomotor, and affective) occur as both content-dependent and content-independent assessments. The assessments most closely associated with a competency would be the content-dependent trait-like assessments.

Marksmanship Competency

To provide an example of how this framework might be used to describe a competency, an example is given below for marksmanship.

Figure 1, presents the psychomotor, affective, and cognitive components of a hypothetical marksmanship competency model. The psychomotor competency includes behaviors known to contribute to accurate shooting such as steady body position and trigger squeeze. The affective domain includes fear of guns. Many people have a fear of guns and it is not hard to see how this would interfere with the development of this competency. Training would be needed to help an individual with a fear of guns overcome that fear in order for that individual to achieve competency at marksmanship. Finally, the cognitive domain includes things like an understanding of how the weapon functions, how to adjust weapon sights, and how to clear a malfunction.

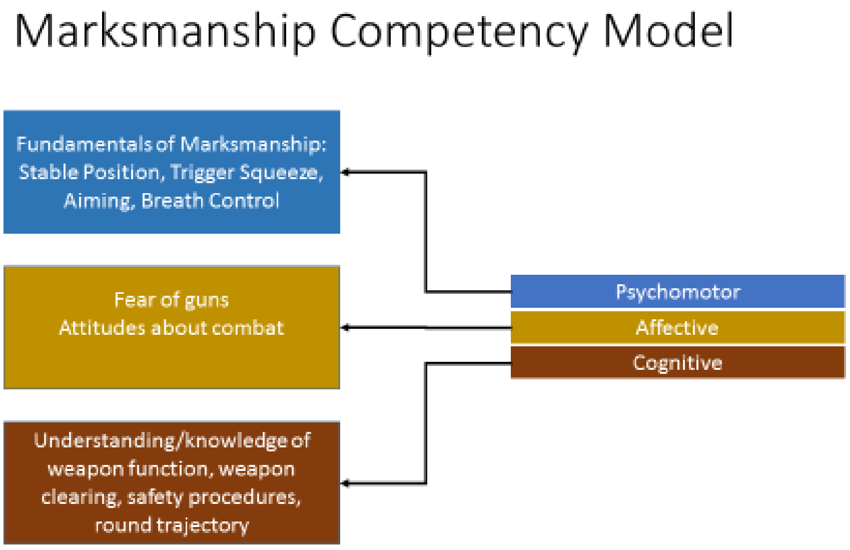

Marksmanship Competency Model
Fundamentals of Marksmanship: Stable Position, Trigger Squeeze, Aiming, Breath Control
Fear of guns
Attitudes about combat
Understanding/knowledge of weapon function, weapon clearing, safety procedures, round trajectory
Psychomotor
Affective
Cognitive

Figure 1. The components of marksmanship competency

Competency Model Framework

Using such a competency model within an adaptive training system would entail defining the levels of the domains of the competency and then developing corresponding measures and training. Table 2 describes this matrix. In this way, a learner's progress in a competency like marksmanship can be modeled using those measures and appropriate training can be delivered to advance the learner through the competency.

Table 2. Competency Framework

Level	Domain		
	Psychomotor	Affective	Cognitive
Level 1	Measures/Training P1	Measures/Training A1	Measures/Training C1
Level 2	Measures/Training P2	Measures/Training A2	Measures/Training C2
Level 3	Measures/Training P3	Measures/Training A3	Measures/Training C3

Level n	Measures/Training Pn	Measures/Training An	Measures/Training Cn

Multiple measures and multiple training options might exist for each level of any domain. Regarding measures, completion of pre-requisite training or prior experience could be used to mark the learner's competency level. Regarding training, it would be up to the domain module author to identify the level to which the training should be associated. Although the table shows equal numbers of levels for each of the three domains of learning, this is not a requirement.

GIFT IN THE WILD

As mentioned above, GIFT can be expected to operate within a larger ecosystem of training systems including live, virtual, constructive, and gaming. For GIFT to be most adaptive, it will need to be able to account for training delivered on those other systems. Likewise, as adaptive training systems become more pervasive throughout that ecosystem, they will likewise need to track the competency of the learner.

An interoperable learner model would provide a means of tracking the learner's competency across multiple training systems. At the present time, few if any systems have the ability to use an interoperable learner model, however, most adaptive training systems generate learner assessments.

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. Figure 2 illustrates how an adaptive training system would use a competency model in this larger ecosystem to provide appropriate training to a learner.

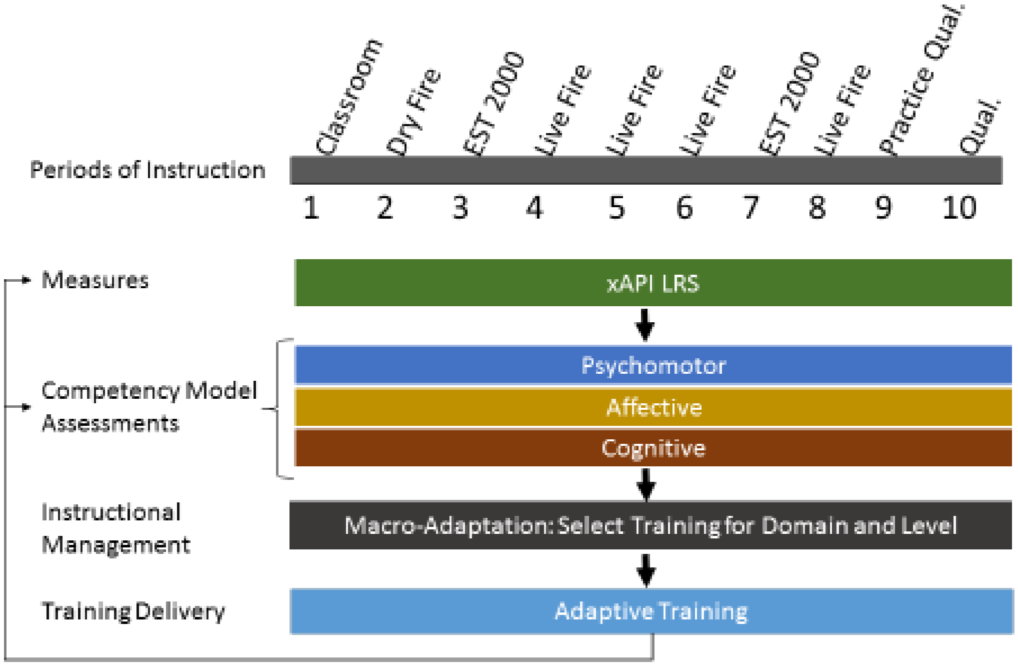


Figure 2. Persistent Learner Competency Model

Figure 2 illustrates how assessments in an xAPI LRS derived from different periods of instruction for basic rifle marksmanship could be used to inform a persistent competency model. In this way an adaptive training system can know where the learner is with respect to the marksmanship competency and this can more effectively drive the macro-adaptation of that adaptive training system.

DEVELOPING GIFT

An opportunity to investigate interoperable learner models in ITSs exists through the integration of ongoing ARL-sponsored efforts. Under the *Support for Training Effectiveness Assessment with Data Interoperability (STEADI)* effort, performance metrics expressed in xAPI statements will be developed for training effectiveness evaluation and to meet the needs of other audiences. In this effort, marksmanship data will be collected using a subset of 4 technology-based training systems. ARL's GIFT already accepts xAPI to an extent. Specifically, GIFT has a minimally functional capability in its Learning Management System (LMS) module to produce and consume simple xAPI data (Hruska, Medford, & Murphy, 2015). A minimally functional macro-adaptive course filtering capability also exists. Extending this integration into the student model enables integration of the findings of multi-faceted interoperable student models into GIFT.

This integration enables the investigation of a number of research questions that have to date been unanswerable given the current state of intelligent tutoring technology. To date, the research into how best to adapt training content based on student performance in intelligent tutoring systems is inconclusive (Durlach & Ray, 2011). Less research has been done investigating best practices in adapting training across domains. For example, if a student performs well in a given curriculum provided in an ITS (e.g., geometry), how would an ITS prescribe an adaptive curriculum in a separate, but related domain (e.g., calculus)? Some unanswered research questions 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,” or 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 ITSs 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, Carroll, Champney, & Padron,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 AND RECOMMENDATIONS FOR FUTURE RESEARCH

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. We envision a multi-faceted learner 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.

As an initial attempt at addressing these issues, future research will integrate interoperable learner models into the GIFT framework using xAPI data. We plan to use a marksmanship use case for our 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 this research is still in its infancy. Consequently, our effort is a first step toward developing definitive guidelines and best practices for how to best leverage interoperable performance data. Further research is needed to expand our 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.

Anderson, L.W., Krathwohl, D.R., Airasian, P.W., Cruikshank, K.A., Mayer, R.E., Pintrich, P.R., Raths, J., Wittrock, M.C. (2001). A Taxonomy for Learning, Teaching, and Assessing: A revision of Bloom's Taxonomy of Educational Objectives. New York: Pearson, Allyn & Bacon.

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.

Clark, R., Chopeta, L. (2004). Graphics for Learning : Proven Guidelines for Planning, Designing, and Evaluating Visuals in Training Materials . Jossey-Bass/Pfeiffer.

Dave, R.H. (1970). Psychomotor levels in *Developing and Writing Behavioral Objectives, pp.20-21*. R.J. Armstrong, ed. Tucson, Arizona: Educational Innovators Press.

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, The 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

Fitts, P. M., & Posner, M. I. (1967). *Human performance*. Belmont, CA: Brooks/Cole.

Gagné, R. M. (1989) *Studies of Learning: 50 Years of Research*. Learning Systems Institute, Tallahassee, FL.

Hruska, M., Medford, A., Murphy, J. (2015). Learning Ecosystems Using the Generalized Intelligent Framework for Tutoring (GIFT) and the Experience API (xAPI). Proceedings of the 17th International Conference on Artificial Intelligence in Education (AIED 2015).

Krathwohl, D.R., Bloom, B.S., Masia, B.B. (1973). Taxonomy of Educational Objectives, the Classification of Educational Goals. Handbook II: Affective Domain. New York: David McKay Co., Inc.

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.

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

Soller, A. 2001. Supporting social interaction in an intelligent collaborative learning system. International Journal of Artificial Intelligence in Education. 12(1):40-62.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.

U.S. Army Training and Doctrine Command. 2011. The United States Army Learning Concept for 2015. TRADOC Pamphlet 525-8-2. Fort Monroe, VA. [Accessed 2015 May] http://www.tradoc.army.mil/tpubs/pams/tp525-8-2.pdf.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

***Gregory Goodwin** is a senior research scientist 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.*

***Jennifer Murphy** is the Chief Executive Officer of Quantum Improvements Consulting, LLC. She has over 10 years of military selection and training research experience, with an emphasis on leveraging innovative technologies for improving training in a measurably effective way. Upon completion of her Ph.D. from the University of Georgia in 2004, Dr. Murphy took a position as a Research Psychologist at the U.S. Army Research Institute for the Behavioral and Social Sciences (ARI). Her research focused on the development of technology-based selection and training measures for cognitive and perceptual skills. Dr. Murphy served as Director of Defense Solutions at Design Interactive, Inc., where she managed a portfolio of training and performance support efforts incorporating cutting edge technology into training solutions for defense clients. Her research has been featured in The New York Times, the Pentagon Channel, Soldier Magazine, and Signal Magazine.*

***Michael Hruska** is a technologist with experiences spanning across standards, emerging technologies, learning, and science. He is a former researcher at the National Institute of Standards and Technology in Gaithersburg, Maryland. He is currently the President / Chief Executive Officer of Problem Solutions, and provides learning technology solutions to government, commercial, and nonprofit organizations. His team has been supporting efforts for the last 6 years at the Advanced Distributed Learning (ADL) Initiative on the future of a Training and Learning Architecture (TLA) and the Experience Application Programming Interface. He holds a Bachelor of Science from the University of Pittsburgh and is a member of the e-Learning Guild, American Society of Training and Development (ASTD) and the National Defense Industrial Association (NDIA).*

Characteristics of a Multi-User Tutoring Architecture

Stephen Gilbert[1], Eliot Winer[1], Joseph Holub[1], Trevor Richardson[1], Michael Dorneich[1], Michael Hoffman[2]
Iowa State University[1], Dignitas Technologies[2]

INTRODUCTION

Intelligent tutor systems have been quite successful in instruction of individuals (Koedinger, Anderson, Hadley, & Mark, 1997; Ritter, Kulikowich, Lei, McGuire, & Morgan, 2007; Vanlehn, et al., 2005), but multiple challenges exist when attempting to tutor a team. Sottilare, Holden, Brawner, and Goldberg (2011) describe some of the architectural challenges of team tutoring at a high level in terms of functional requirements. In this paper we describe specific challenges in terms of implementing a team architecture within the Generalized Intelligent Framework for Tutoring (GIFT), including simultaneous startup and synchronization with distributed team members, maintaining state of multiple users, and timing feedback for teams and individuals appropriately.

Illustrative Example: The Recon Task

To provide an example that drives functional requirements for simple team tutoring, we present the Recon Task. In this reconnaissance mission, a team of two soldiers (Alpha Team, made up of Alice and Bob) is responsible for conducting surveillance over respective sectors of a specific area. Each soldier has three responsibilities, or subtasks, in this Recon Task: 1) identify opposing forces (OPFOR) within the sector, 2) report to the teammate if an OPFOR is moving into the teammate's sector, and 3) acknowledge the alert if teammate reports an incoming OPFOR. This task is described in further detail as a broad experimental test bed for teams by Bonner et al. (2015). See Figure 1. We implemented the Recon Task scenario in VBS2, since that game engine was compatible with the current GIFT release. For ease of implementation, all three subtasks are accomplished by typing individual keyboard keys, e.g. "Type Q to identify an OPFOR in your sector."

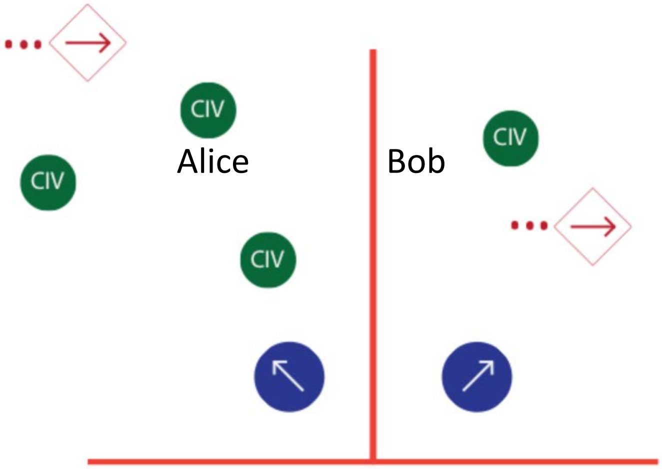


Figure 1: The Recon Task. Alpha Team members Alice and Bob in blue must scan their sectors for opposing forces (diamonds) and alert each other if one is moving into the partner's sector. Civilians are distracters.

To provide team tutoring for the Recon Task, we would like the team tutor to be able to offer feedback both to each individual and to the entire team, depending on the dynamics of its performance. For simplicity's sake, let us assume initially that feedback is given in real-time based on errors in any of the three subtasks, rather than other varieties of feedback, such as real-time feedback that praises good performance, prompts that remind members of required actions ahead of time, or summative feedback in an after-action review. This approach leads to the possible feedback messages shown in Table 1. We assume that these feedback messages will be given to individuals by GIFT, and that each team member is logged into an individual computer running VBS2 and GIFT. We will assume that the decision of whether to address the feedback to an individual or to the team is simple: if both individuals' performances merit the same feedback within a very close time window, the feedback is addressed to "Alpha Team."

Let us also assume that our GIFT team tutor will maintain learner models for Alice, Bob, and the Alpha Team as a whole based on performance on these subtasks, essentially keeping score. In Table 1, to illustrate that calculation methods of team performance can vary, the team performance for Subtask 1 is the average of the team members' performance, while for Subtasks 2 and 3, the team performance is the minimum of the both team members' performance.

Table 1: Feedback for each subtask of the Recon Task, as well as sample performance scores for Alice, Bob, and the Alpha Team as a whole. Each performance column can be considered to be a simple learner model.

Subtask	Individual Feedback	Team Feedback	Alice Perf.	Bob Perf.	Alpha Team Perf.
1. Identify OPFOR	Alice/Bob, identify OPFOR as quickly as possible.	Alpha Team, identify OPFOR as quickly as possible.	80%	50%	65%
2. Alert of incoming.	Alice/Bob, communicate crossings promptly.	Alpha Team, communicate crossings promptly.	40%	60%	40%
3. Acknowledge alert.	Alice/Bob, acknowledge all alerts.	Alpha Team, acknowledge all alerts.	100 %	90%	90%

This scenario now provides sufficient requirements that we can describe the technical architecture and its corresponding challenges. Before discussing our architecture, we briefly describe previous efforts at team tutoring to explore whether previous architectures could support a task like this.

OTHER EFFORTS IN TEAM TUTORING

In the 1990s, the Advanced Embedded Training System (AETS) was designed to facilitate team-based training a Naval Air Defense Team (Zachary, et al., 1999). AETS featured instruction for individual operators based on tracking keystrokes, speech, and eye movements and a comparison of operator behavior with expected behavior. It updated student skill models throughout the simulation. The AETS also monitored individual and team performance and provided a dashboard of relevant information to human trainers, but did not have an architecture in place itself to offer automated team feedback. Another team led by Zachary (Zachary, Santarelli, Lyons, Bergondy, & Johnston, 2001; Zachary, Weiland, Scolaro, Scolaro, & Santarelli, 2002) later created SCOTT, a system for operational team training that used synthesized teammates. However, this system also did not offer automated team feedback.

Marsella and Johnson (1998) did offer feedback regarding team behavior using PuppetMaster, an agent that monitored activities throughout a large scale simulation environment and aggregating feedback and guidance for a human instructor. The researchers in this case decided that it was impossible to do classic intelligent tutor knowledge tracing (inferring an agent's plan based on known possible plans to a goal), so instead they tracked agents' behaviors to see whether they aligned with current goals. Their agent-based approach may be a useful inspiration for our own architecture, though they concluded with a desire for a method of integrating a model of an individual agent's behavior with a model of team behavior.

Rickel and Johnson (1999) used avatars in a virtual environment for training. Their virtual instructor, Steve (Soar Training Expert for Virtual Environments) had originally been developed for individual training, but in the 1999 research was adapted for team training. The authors describe similar requirements to our team tutoring task. A Steve agent must be able to track multiple other entities in the virtual environment (other Steve agents or people) and direct communication to appropriate entities for team coordination. Steve agents were implemented using Soar (Laird, Newell, & Rosenbloom, 1987) and thus had modules for perception, cognition, and motor control. The perception module monitors the full simulation state (including other entities). To enable Steve's task-based plans and goals to accommodate team training, the researchers made the task descriptions more modular, enabling steps to be done in variable order when possible, and thus be done in parallel or in a non-specified order by different team members. Task steps were mapped to roles that other team members could take on. By focusing on a hierarchical list of task steps, the agents could have a basic do-everything-myself plan but regularly check the state of the simulation to see what is already done and what remains. The researchers were also able to use this task-based architecture to enable two virtual instruction agents to speak with each other.

Nair, Tambe, Marsella and Raines (2004) explored team behavior by creating agents that analyzed team behavior in a soccer context. Their automatic team analyst, ISAAC, much like our architecture, contained multiple models of behavior, a model for individual agents, a model for multiple agents (e.g., two soccer players), and a model for the entire team. The multiple agent model focused on recognizing specific patterns of soccer play among small numbers of players. This multiple agent model might also be a possible inspiration for our own team architecture.

TEAM ARCHITECTURE

To create a tutor for the team-based Recon Task, the architecture first requires a representation of each participant's system: a computer running the simulation (VBS2) and a tutor client that can 1) monitor what the participants are doing and 2) give the participants tutor feedback when needed. That tutor client in our case is the GIFT Gateway Module, running as a local server on each participant's computer, and displaying its feedback for each participant in a webpage. See Figure 2. The Gateway Module also uses two plugins to 1) translate VBS2 DIS messages into a format for GIFT and 2) exchange commands with VBS2 via its API.

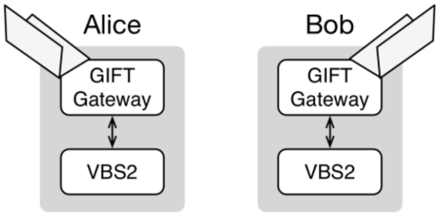


Figure 2: The architecture of each team member's computer, running VBS2 and the GIFT Gateway module.

Because the Recon Task requires evaluation of both team and individual performance, some information must be stored in common across all participants in a separate layer. In specifying this common layer, we consider two types of information: information updates that are used short-term (e.g., DIS packets, simulation events, and learner actions) and long-term stored information (e.g., individuals' skills demonstrated over time, and the set of conditions and feedback that are given based on performance in the task domain). Because of the short-term updates and longer-term storage that must be shared across team members to enable this tutor, the architecture requires both a method of communicating update messages and a method of maintaining information over time. In our implementation, the primary GIFT server provides the common layer for some of both. GIFT uses a third-party messaging system called ActiveMQ for communication updates, and the long-term storage occurs within GIFT's Domain Module and its Learner Module.

There are two kinds of short-term information updates that need processing. ActiveMQ is used within GIFT to communicate within the modules, and in particular, to pass learners' actions to the modules. If the updates are of interest to a module, e.g., if the message is about Alice identifying an OPFOR, the tutor will process it. Higher fidelity information that needs to be communicated at higher frequency between players, e.g., game engine state changes such as "Alice just moved to x, y, z, so draw her at new position a, b, c on Bob's screen," are handled by the VBS2 multi-player module. It is worth noting that if another client simulation were used for team tutoring that did not have a similar multi-player module to handle high fidelity synchronization, this architecture would need to change somewhat.

Two separate modules, the Domain Module and Learner Module, store information longer-term. The Domain Module contains preprogrammed feedback that the learners will receive, along with the conditions that trigger those feedback messages. Those conditions are passed to the Pedagogical Module during initialization of the scenario for execution during run-time. The Learner Module contains accumulated skill ratings for the learners and the team, and its information is updated frequently based on team performance. See Figure 3.

In our example, the Learner Module stores the information in the three blue columns at the far right of Table 1, the performance scores for Alice, Bob, and Alpha Team for the three subtasks within the Recon Task. The Domain Module stores the feedback that is shown in the light orange second and third columns. The conditions for the subtask "Identify OPFOR," for example, include "A learner who identifies an OPFOR after more than 10 seconds of its appearing on screen is graded *Below Expectation*" and "A learner who identifies sooner that 5 seconds after its appearance is *Above Expectation*." The Domain Module also contains conditions for the team, e.g., for the Acknowledge Alert subtask, a condition might be "If both players perform Below Expectation, then the team is graded *Below Expectation*." The Domain Module also references the evaluation algorithms for assessment of different learning goals, sometimes called check functions, which are typically java classes customized based on the type of assessment needed for a specific scenario. In our Recon Task, for example, the check functions evaluate whether OPFORs have been identified on time, whether alerts of incoming OPFOR have been given and when, and whether alerts have been acknowledged. While the Domain Module steadily receives update messages and sends back replies according to the knowledge stored within it, its knowledge does not change within a given scenario. In a more complex tutor, the feedback messages themselves might be variable, dynamically adapting to the actions of the learner.

To walk through a typical communication flow within our Recon Task example (the "tutor loop"), the learner takes an action within VBS2, and VBS2 sends the game state to the Gateway. The Gateway translates the VBS2 game state to a GIFT message and sends it to the ActiveMQ bus. Now see the dotted line arrows and numbers in Figure 3. The Domain Module retrieves the message, processes it, and when appropriate, outputs an assessment of a learner's performance for the Learner Module (1). The Learner Module compares the new performance with the current performance, and if the learner state changes, it passes a message to the Pedagogical Module (2) via the message bus. That module decides whether a pedagogical intervention is warranted. If yes, it passes a message to the Domain Module (3) via the message bus. The Domain Module then determines the particular tactic for implementing feedback, e.g., displaying a message in the GIFT webpage or in VBS2 itself, and passes that along the message bus to the Gateway.

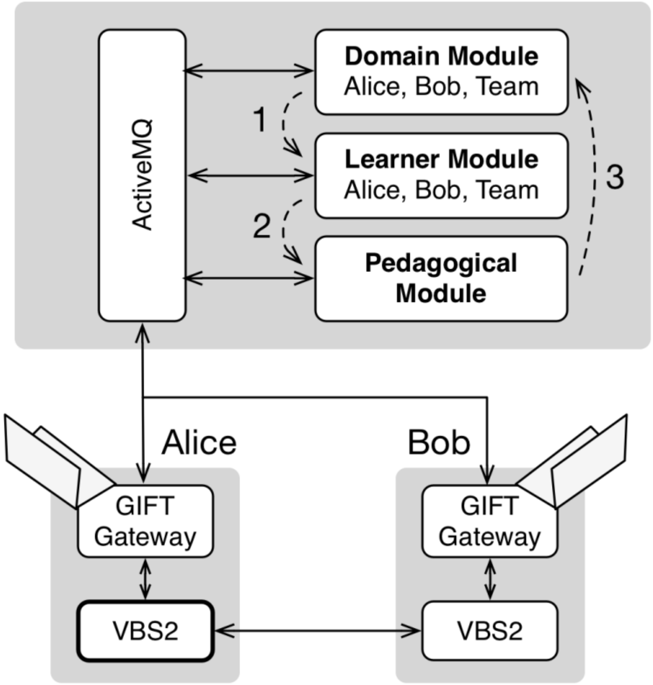


Figure 3: The GIFT team architecture that supports the Recon Task. The steps of the "tutor loop" are numbered.

What Makes This a Team Tutor Architecture?

The architecture described so far is not significantly dissimilar from a typical intelligent tutoring architecture in which there is a learner client interface, a tutor running on a separate server, and a translation module in the middle that sends learner actions to the tutor and receives feedback from the tutor to display to the user in the client (Cheikes, et al., 1999; Ritter & Koedinger, 1996). However, two main challenges arise in setting up the team tutor that must still be answered. 1) How is the startup of the scenario handled with multiple users? 2) How does the tutor manage feedback to individuals vs. feedback to the team?

Management of Multi-user Startup

Currently in GIFT, and in the architecture described above so far, multiple users are supported, but each with his or her own GIFT session. The sessions are independent and not aware of each other, however, so that there is no communication across sessions to promote team activities. To link the individual participants' sessions, a team session is required for synchronization at startup. This team session identifies participants and sets up the communication channels so that individuals can receive feedback specific to them, and they can all receive team feedback if appropriate. The solution we have implemented for this is a GIFT Network Lobby. Just as in other multiplayer games, in which users login and then wait for other players to join before entering the game, learners choose a team-based scenario within GIFT, choose a role they will play in the scenario, and then wait for other roles to be filled before being able to launch. In our particular implementation of the team tutor for the Recon Task, this architecture leads to the startup procedure described in Table 2.

Table 2: Sequences of actions by two players at startup that illustrate synchronization using the GIFT Network Lobby. Alice starts first.

Alice	**Bob**
Launches VBS2. (First launch becomes the master VBS2 node.)	Launches VBS2.
Launches GIFT webpage & logs in.	Launches GIFT webpage & logs in.
Chooses Recon Task and Player 1 role. (Enters GIFT Network Lobby; waits for Player 2.)	Chooses Recon Task and Player 2 role.

All players present; GIFT Network Lobby launches VBS2 Recon Task Scenario via VBS2 plugin.

Management of Feedback for Both Team and Individuals

This element of team tutoring is the most complex; it can be difficult to coordinate the mechanisms offering feedback to the team and to the individual so that they do not conflict or overlap. In traditional individual tutor, there is one player receiving feedback and one tutor generating it. This conflict can happen on the back end, within the tutor, if the tutor's conditions indicate that a player currently merits feedback both as an individual and as a team member. This issue of prioritizing multiple feedback messages to give can happen within an individual tutor as well, but that issue can be resolved with priorities assigned to the conditions, done by Le, Menzel, and Pinkwart. (2009) among others. In the constraint-based tutor, ASPIRE, Mitrovic et al. (2009) organized conditions by domain concept to address this issue. However, if the components in a team tutor that give feedback are created as semi-autonomous processes, e.g., a feedback agent for the individual, and a separate feedback agent for the team, then simple prioritization does not work as well; the challenge becomes more of coordination issue.

This issue of multiple conflicting messages can also arise on the front end, in the learner's user interface. If the tutor is allowed to give the learner multiple messages, a decision must be made as to how they will be timed and visually arrayed or played back so that one message does not get upstaged by the other(s).

In our case, because part of our goal was to extend GIFT to accommodate team tutoring, we used its existing Domain Knowledge File (DKF) format. A DKF file typically stores the conditions and feedback for an individual tutor. One approach we considered was to use one DKF for the individual tutoring, and one DKF file for the team tutoring. However, since our requirements for the Recon Task include having the individual feedback include the participant's name (e.g., "Alice, identify…"), and current DKF feedback statements cannot be customized, we settled on using three DKFs, one for Alice, one for Bob, and one for the team. This implementation has led to some duplication of code, in that the same conditions and feedback are repeated in Alice's DFK and Bob's DKF. Also, because some of the conditions (but not all) overlap with the Team DKF, there is additional code redundancy. For example, if we want to check whether Alice is acknowledging alerts, we do that in her DKF. We would also have similar condition code in Bob's DKF to measure whether he is acknowledging alerts. Finally, we might have a slight variation on that condition in the Team DKF to measure whether the team as a whole acknowledging alerts. This approach made us realize how helpful it would be in the longer term to have an architecture that allowed domain knowledge inheritance or nesting of files. E.g., there could be a single individual DKF class that spawns customized instances of individual DKFs for each learner. Additionally, the Team DKF could allow inclusion of or referencing of conditions and feedback from the individual DKF class to eliminate that redundancy.

This current implementation also has the issue that because the three DKF files are independent, it is not possible to create conditions that depend on the actions of multiple players (or Boolean combinations of conditions). For example, I could not write a condition like, "If Alice is doing well at identification, AND Bob is not, then tell Alice…" Also, because the DKF files are independent, an individual learner may receive multiple messages, one from the individual DKF and one from the team DKF. In our current implementation we do not have a method of prioritizing these according to pedagogy or type of condition. We could implement a system-wide rule such as, "If there are feedback messages from both team and individual DKFs, give the team one." Currently our implementation allows both messages to appear on screen.

One potential approach to addressing the above issues might be to have scripting language statements allowed in both conditions and feedback. If the feedback message could be written, "{name of learner}, identify…" and the message format sent by ActiveMQ included the learner's ID, then multiple individual DKFs would not be needed. To address multiple Boolean conditions, an author might write a condition that translates to "If {learner 1} is doing well at identification, AND {any other learner} is not, then tell {learner 1} …" Writing conditions that depend on time and learner history could also be facilitated by writing a condition that includes relative comparisons instead of absolute ones, such as, "If performance level of {any learner} has improved more than one level in the past 5 minutes, then…" Such a scripting language could be called GIFTscript, and be based on the principles of Applescript (2007) and Tutorscript (Blessing, Gilbert, & Ritter, 2006), a language used with Cognitive Tutors at Carnegie Learning, Inc.

CONCLUSIONS

When tutoring teams, we are primarily interested in two scenarios: teams that are co-located and synchronous, and those that are distributed and asynchronous. In its current state, multiple GIFT instances can be connected via a local network. By connecting trainees via a multiplayer training application, we can simulate the co-located, synchronous scenario. However, limitations exist within the current architecture: it is difficult to provide the complex tutoring to both teams and individual members that a human coach would provide because the GIFT DKF models (one for each team member and one for the team) remain static and independent of each other. This paper describes an initial attempt to create an architecture to support team tutoring. While we have been successful in doing so, an understanding of the lack of scalability of this architecture and its required simplicity will inform future development of a more robust future team tutoring architecture within GIFT.

ACKNOWLEDGEMENTS

The research described herein has been sponsored by the U.S. Army Research Laboratory - Human Research & Engineering Directorate (ARL-HRED). 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

(2007). Introduction to AppleScript Overview Retrieved July 6, 2012, from http://developer.apple.com/applescript/

Blessing, S., Gilbert, S. B., & Ritter, S. (2006). *Developing an authoring system for cognitive models within commercial-quality ITSs.* Paper presented at the Nineteenth International FLAIRS Conference.

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 *Proceedings of the Workshops at AIED 2015*. Madrid, Spain.

Cheikes, B. A., Geier, M., Hyland, R., Linton, F., Rodi, L., & Schaefer, H.-P. (1999). Embedded training for complex information systems. *International Journal for Artificial Intelligence in Education, 10*, 314-334.

Koedinger, K. R., Anderson, J.R., Hadley, W.H., & Mark, M.A. (1997). Intelligent tutoring goes to school in the big city. *International Journal for Artificial Intelligence in Education, 8,* 30-43.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial intelligence, 33*(1), 1-64.

Le, N.-T., Menzel, W., & Pinkwart, N. (2009). Evaluation of a constraint-based homework assistance system for logic programming. In S. C. Kong, H. Ogata, H. C. Arnseth, C. K. K. Chan, T. Hirashima, F. Klett, J. H. M. Lee, C. C. Liu, C. K. Looi, M. Milrad, A. Mitrovic, K. Nakabayashi, S. L. Wong & S. J. H. Yang (Eds.), *Proceedings of the 17th International Conference on Computers in Education*. Hong-Kong: Asia-Pacific Society for Computers in Education.

Marsella, S. C., & Johnson, W. L. (1998). *An instructor's assistant for team-training in dynamic multi-agent virtual worlds.* Paper presented at the Intelligent Tutoring Systems.

Mitrovic, A., Martin, B., Suraweera, P., Zakharov, K., Milik, N., Holland, J., et al. (2009). ASPIRE: an authoring system and deployment environment for constraint-based tutors. *International Journal of Artificial Intelligence in Education, 19*(2), 155-188.

Nair, R., Tambe, M., Marsella, S., & Raines, T. (2004). Automated assistants for analyzing team behaviors. *Autonomous Agents and Multi-Agent Systems, 8*(1), 69-111.

Rickel, J., & Johnson, W. L. (1999). *Virtual humans for team training in virtual reality.* Paper presented at the Proceedings of the ninth international conference on artificial intelligence in education.

Ritter, S., & Koedinger, K. R. (1996). An architecture for plug-in tutor agents. *Journal of Artificial Intelligence in Education, 7*(3), 315-347.

Ritter, S., Kulikowich, J., Lei, P., McGuire, C. L., & Morgan, P. (2007). What evidence matters? A randomized field trial of Cognitive Tutor Algebra I. In T. Hirashima, U. Hoppe & S. S. Young (Eds.), *Supporting Learning Flow through Integrative Technologies* (Vol. 162, pp. 13-20). Amsterdam: IOS Press.

Sottilare, R., Holden, H. K., Brawner, K. W., & Goldberg, B. S. (2011). *Challenges and Emerging Concepts in the Development of Adaptive, Computer-based Tutoring Systems for Team Training.* Paper presented at the The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), Orlando, FL.

Vanlehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R., Taylor, L., et al. (2005). The Andes Physics Tutoring System: Lessons Learned. *Int. J. Artif. Intell. Ed., 15*(3), 147-204.

Zachary, W., Cannon-Bowers, J., Bilazarian, P., Krecker, D., Lardieri, P., & Burns, J. (1999). The Advanced Embedded Training System (AETS): An Intelligent Embedded Tutoring System for Tactical Team Training. *International Journal for Artificial Intelligence in Education, 10*, 257-277.

Zachary, W., Santarelli, T., Lyons, D., Bergondy, M., & Johnston, J. (2001). Using a community of intelligent synthetic entities to support operational team training: DTIC Document.

Zachary, W., Weiland, W., Scolaro, D., Scolaro, J., & Santarelli, T. (2002). *Instructorless team training using synthetic teammates and instructors.* Paper presented at the Proceedings of the Human Factors and Ergonomics Society Annual Meeting.

ABOUT THE AUTHORS

Stephen Gilbert, Ph.D*., is an associate director of the Virtual Reality Applications Center and assistant professor of Industrial and Manufacturing Systems Engineering at Iowa State University. His background includes cognitive science, engineering, and Human Computer Interaction. He is PI on the team tutoring project with ARL HRED and also co-led the development of a reconfigurable mixed-reality training environment for the warfighter. Dr. Gilbert has over 10 years' experience developing intelligent tutoring systems.*

Eliot Winer, Ph.D*., is an associate director of the Virtual Reality Applications Center (VRAC), associate professor of mechanical engineering, and a faculty affiliate of the human computer interaction (HCI) graduate program at Iowa State University. He has integrated four virtual and three live environments in a simultaneous capability demonstration for the Air Force Office of Scientific Research and has co-led the development of a next-generation mixed-reality virtual and constructive training environment for ARL HRED. Dr. Winer has over 15 years' experience with virtual reality, computer graphics, and simulation technologies.*

Joseph Holub, MS *is a graduate research assistant at the Virtual Reality Applications Center (VRAC) at Iowa State University where he is finishing his Ph.D. in human computer interaction (HCI) and computer engineering. He has worked on projects for path planning of unmanned aerial vehicles, live virtual constructive training of dismounted soldier, and visualization of large data using contextual self-organizing maps. Currently, he is working on building tools for augmented reality assembly in manufacturing as well as the GIFT team training architecture. His Ph.D. research is on visualizing functional imaging data on multiple hardware platforms.*

Trevor Richardson, MS *is a graduate research assistant at the Virtual Reality Applications Center (VRAC) at Iowa State University where he is finishing his Ph.D. in human computer interaction (HCI) and computer engineering. He has worked on augmented reality assembly in manufacturing as well as the GIFT team training architecture. His Ph.D. research is on optimization techniques using contextual self-organizing maps.*

Michael Dorneich, Ph.D., *is associate professor of Industrial and Manufacturing Systems Engineering and a faculty affiliate of the human computer interaction (HCI) graduate program at Iowa State University. Dr. Dorneich's research interests focus on creating joint human-machine systems that enable people to be effective in the complex and often stressful environments found in aviation, robotic, learning, and space applications. Dr. Dorneich has over 19 years' experience developing adaptive systems which can provide assistance tailored to the user's current cognitive state, situation, and environment.*

Michael Hoffman, MS *is a Senior Software Engineer at Dignitas Technologies with a M.S. in Computer Science from the University of Central Florida and over 10 years of experience in software development. He has worked on various efforts in the fields of modeling and simulation as well as integrating numerous software and hardware systems such as third party simulations and sensors. Michael is the lead on ARL's Generalized Intelligent Framework for Tutoring (GIFT) project.*

The Unpacking of Team Models in GIFT

C. Shawn Burke[1], Jennifer Feitosa[1], & Eduardo Salas[1]
University of Central Florida[1]

INTRODUCTION

The use of computer based tutoring programs for learning has recently seen a renewed interest in the social sciences, particularly in terms of utilizing such systems for the training and development of teams (Sottilare, Holden, Brawner, & Goldberg, 2011). Researchers have begun to push for an intelligent tutoring system (ITS) for individuals and teams which can be adapted to multiple training situations and scenarios. Such a push has produced the Generalized Intelligent Framework for Tutoring (GIFT; Sottilare et al., 2011). While this framework provides a starting point for a more advanced and flexible intelligent tutoring process, additional research is necessary to realize the potential of GIFT as a team tutoring system. Although the teamwork literature has received ample scholarly attention (Cannon-Bowers & Bowers, 2010), little remains known about the real ontology of core team behaviors, attitudes, and cognition and their consequences to team outcomes. Drawing from Sottilare et al. (2011) team model and others that tried to synthesize the teamwork and team training in a qualitative way (e.g., Campion, Medsker, & Higgs, 1993; Cannon-Bowers & Bowers, 2010; Dyer, 1984; Klein et al., 2009; Salas et al., 1992, 2008; Smith-Jentsch et al., 1998, 2008), we extract key variables to scrutinize in a quantitative manner. Furthermore, the relationship between these variables is complex, considering the different features and individual characteristics, intervention design, and environmental variables involved in team training.

While initial team models have been theorized, further work is needed to identify a complete design architecture (e.g., delineating specific GIFT team model components, behavioral markers and metrics). This design architecture must be rooted in principles of intelligent tutoring, but also be based upon the science behind teamwork/team performance. This paper describes the first step in such an effort. Specifically, it describes the process and results of a quantitative synthesis of the existing science to inform the refinement of the team models. We systematically examine the empirical work on teams to synthesize the literature. For this, we conducted meta-analyses on the relationships between specific team behaviors, attitudes, and cognition with specific team outcomes to produce a refined set of team state models (e.g., which constructs are important to include in specific models).

Methodology

Meta-analytic coding procedures were utilized to examine empirical articles describing the behavioral, cognitive, and affective antecedents to team outcomes (i.e., performance, learning, satisfaction, and viability).

Literature Search

To identify primary studies for inclusion in the meta-analyses, a search was conducted using the American Psychological Association's PsycINFO (2003-July 2013), Defense Technical Information

Center, and ProQuest for combinations and variations of the following keywords: performance/ competency/ trust/ cognition/ affect/ communication/ intelligent tutoring/ human-computer interaction/ virtual human/ mood/ emotion/ skill/ knowledge/ ability/ responsibilities/ roles/ distributed/ virtual/ after action review/ feedback/ leadership/ cohesion/ personality/ effectiveness; paired with either team/ unit/ group/ squad/crew. Furthermore, the following were used as secondary search terms: progress/ goals/ experience/ perceptions/ engagement/ boredom/ confusion/ frustration/ situation awareness/ training/ coordination/ collaboration/ motivation/ cohesion/ learning/ leadership/ training/ building monitoring/ goal setting/ instructional strategies/ debriefing/ decision making/ event-based training/ mental models (team, shared)/ processes/ shared cognition/ simulation based training/ development/ transactive memory systems/ backup behavior/ planning/ coordination/ action/ transition. Additionally, snowball, backtracing approaches, and additional searches that included 'team and learning'/ 'teams and satisfaction'/ 'teams and viability'/ 'teams and performance' were used to supplement our searches.

In searching for primary studies, the search was bounded to include only those articles published/written during the 2003-2013 time period. The time frame chosen was chosen to supplement a previously conducted literature search done by UCF on similar topics that covered the period of 1984-2003. Additionally, this was done to make the model meaningful to current organizations (given the degree to which the nature of work has changed over the past 10 years) and to complement and extend a number of meta-analyses were published during the early 2000s (e.g., leadership by Burke et al., 2006; cohesion by Beal et al., 2003; team conflict by DeDreu & Weingart, 2003; etc.). Our initial searches yielded 5991 unique articles.

Inclusion Criteria

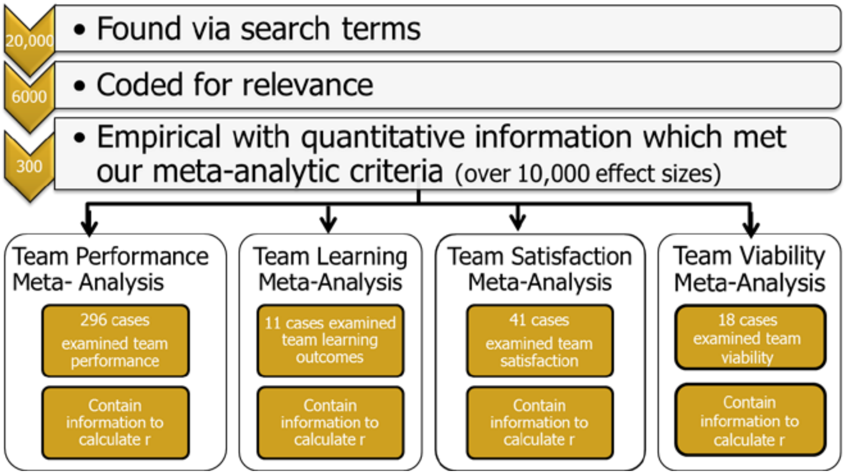

20,000
• Found via search terms
6000
• Coded for relevance
300
• Empirical with quantitative information which met our meta-analytic criteria (over 10,000 effect sizes)
Team Performance Meta- Analysis
296 cases examined team performance
Contain information to calculate r
Team Learning Meta-Analysis
11 cases examined team learning outcomes
Contain information to calculate r
Team Satisfaction Meta-Analysis
41 cases examined team satisfaction
Contain information to calculate r
Team Viability Meta-Analysis
18 cases examined team viability
Contain information to calculate r

Figure 1. Filtering Methodology

To be coded and included in analyses, articles needed to meet a number of boundary conditions (requirements). First, the study had to contain enough information to calculate a correlation between the team variables to be included in our analysis. Second, the focus of the article must be on teams whose members were interdependent. Third, with respect to team size, teams which exceeded 9 people were not included due to a desire to focus on team performance within small teams. Finally, top management teams were excluded due to the unique nature of these teams. The use of the above inclusion criteria resulted in a final meta-analytic database of approximately 300 primary studies. This further broke down into approximately 296 examining team performance as an outcome, 11 with team learning, 41 with team satisfaction and 18 with team viability as an outcome (see Figure 1 for breakdown of results). This resulted in over 10,000 effect sizes prior to composites being created.

Coding Procedure

Studies that passed the inclusion criteria were coded on several categories, including sample characteristics, reliability of measures, and effect sizes. To facilitate the quantitative coding process, a codebook was developed which detailed all of the components of the coding scheme. Prior to beginning the actual coding each coder attended a team agreement meeting to ensure that the first 50 articles coded were consistent across coders in an effort to maintain inter-coder reliability. Each coder also received effect size and composite calculation training from a UCF faculty member with expertise in that area. Subsequently, each week, pairs of coders were assigned articles whereby they came to consensus on which articles were deemed to be codeable (based on the boundary conditions specified earlier). Next, each article was coded by each individual in the pair and any discrepancies were resolved through a consensus meeting. To facilitate coding towards the end of dataset, each pair of raters would come to consensus on codeability, but then split those articles in half so that each individual coded one half of the identified articles.

Analyses

For the quantitative analysis, we followed the Hunter and Schmidt (2004) guidelines for a random-effects meta-analysis of correlations. When multiple effect sizes were presented within a single sample, composites were created (Nunnally, 1978), and if the information required to calculate a composite was not available, the mean of the effect sizes was used. In cases where a composite or average was calculated, the reported reliability estimates were used in the Spearman-Brown formula in order to calculate the reliability of the composite or average. The calculation of the composite correlations and all analyses were performed using SAS Enterprise Guide 6.1 and SAS macros that executed original syntax as well as syntax modified from Arthur, Bennett, and Huffcut (2001). Our results included a sample-weighted mean point estimate of the study correlations (r) as well as the corresponding 95% confidence interval (which expresses the amount of error in r that is due to sampling error and is used for statistical significance testing). We also include the number of independent samples (k) and cumulative sample size (N) included in the calculation of the correlation (r_c) after correcting for unreliability in the predictor and criterion. Corrections for unreliability were performed using only the reliabilities reported in each article.

Results

Team Performance

Team performance is one of the main constructs in teams research (e.g., Bell, 2007; Cannon-Bowers & Bowers, 2010). It has been defined as “the extent to which the productive output of a team meets or exceeds the performance standards of those who review and/or receive the output” (Hackman, 1987, p. 323). For the purposes of this review, team performance was characterized as the evaluation of the outcomes of team processes relative to some set of criteria. A judgment of how well the results of teamwork meet some set of standards (objective or subjective). According to our meta-analytic findings, team behaviors explain up to 42% of the variance in team performance. Considering the importance of distinguishing specific behaviors, we highlight action processes and organizational citizenship behaviors (OCBs) as the most important. These were followed by communication (13%), coordination (i.e., mutual support, 16%; reflexivity, 14%), leadership (11-17%), conflict management, transition processes, and conflict. A number of studies were found in this area of research, giving us the confidence regarding the moderate explanatory power of team behaviors to team performance. Similarly, a wealth of research exists linking attitudes to team performance. The most prominent attitudes were collective efficacy and psychological safety (17-20%), followed by trust and cohesion (9-15%). Beyond behaviors and attitudes, the examination of team cognition showed significant influence to team performance. Transactive memory systems and shared mental models accounted for 20% and 10% of the variance in team performance, respectively. Surprisingly, but with a caveat, situational awareness was the one construct to show the largest relationship with performance. However, this conclusion is based on small sample size that calls for further investigation in order to strengthen confidence in the findings. Table 1 below shows the available intercorrelations across the team constructs that influence performance.

Table 1. Team Performance Correlation Matrix

		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1. Communication									
2. Coordination		**0.46**							
	k	15							
	N	1123							
3. Conflict		**-0.19**	**0.35**						
	k	10	3						
	N	627	262						
4. Leadership		**0.31**	**0.45**	**0.10**					
	k	11	11	2					
	N	789	947	36					
5. Trust		**0.79**	**0.03**	**-0.74**	**0.68**				
	k	5	6	1	6				
	N	186	556	67	409				
6. Collective efficacy		**0.45**	**0.49**	**-0.43**	**0.43**	**0.73**			
	k	1	3	2	6	2			
	N	102	341	113	774	240			
7. Cohesion		**0.64**	**0.52**	**-0.41**	**0.57**	**0.81**	**0.68**		
	k	5	8	6	5	1	7		
	N	472	719	415	324	36	382		
8. Performance		**0.36**	**0.30**	**-0.17**	**0.33**	**0.30**	**0.45**	**0.39**	
	k	64	49	31	46	23	27	23	
	N	4183	3805	2062	3839	1617	2657	1540	

Note. Harmonic mean= 1173

Team Learning

For the purposes of this review, team learning has been defined as the acquisition of knowledge or skills through experience, practice, study, or by being taught. It is important to understand what antecedents can foster more learning, which has been highlighted as a core objective for any training intervention (Mesmer-Magnus & Viswesvaran, 2010). According to our meta-analytic results, team behaviors account for 7-36% of the variance in team learning. Specifically, conflict and conflict management appear as important antecedents, but the amount of studies available in this area limits the interpretation of such finding. Communication and reflexivity appear as main antecedents with a higher number of included studies, and accounting for 25% and 15% of the variance respectively. Earlier work also suggests coaching/leadership can play a significant role in learning. Not surprisingly, attitudes reflective of cooperation account for 37% of the variance in team learning. The role of psychological safety (74%), cohesion (44%), and trust (27%) becomes evident, but the issue of number of studies remains. The findings regarding team learning show a promising avenue that calls for future research to strengthen the confidence in the findings.

Team Satisfaction

Team satisfaction refers to the extent to which members feel content to be part of the team (Hackman, 1987). This evaluation of whether or not team members are satisfied with their team participation plays a central role in determining team effectiveness. In an effort to increase these positive outcomes, team processes have often been examined as a way to increase in team satisfaction (LePine, Piccolo, Jackson, Mathieu, & Saul, 2008; Marks, Mathieu, & Zaccaro, 2001). Correspondingly, our results showed that team behaviors were the most studied antecedent of team satisfaction. Team behaviors accounted for 44% of the variance, highlighting leadership, reflexivity, conflict, coordination, and conflict management (28-42%) as key antecedents. It is important to point out that different aspects of leadership account for different amounts of variance in satisfaction. Additionally, mutual support and conflict accounted for moderate amounts of variance. In regards to attitudes, trust, collective efficacy and cohesion accounted for the most variance in team satisfaction (42-77%). Even though those amounts of variance were significant, the number of studies which included these variables were small. Similarly, when making conclusions about team cognition and team satisfaction, the 23% that team cognition as a group accounted for is limited to no situational awareness studies, and 1 or 2 studies including either transactive memory and team mental model constructs.

Team Viability

Team viability refers to the desire to remain a part of the same team for future performance episodes. This is an especially important outcome when the interaction between teammates and reliance on one another is a reoccurring one. Surprisingly, team behaviors –as a group– account for little variance in team viability. The construct that accounted for 19% and the most variance was conflict, followed by mutual support and coordination. Attitudes, on the other hand, often showed moderate to high explanatory variance of team viability, ranging from 29% to 72%. Collective efficacy was the most important attitude, followed by cohesion. In regards to cognition, transactive memory systems were the only variable linked to team viability in the literature. When considering this team outcome, almost every effect had to be looked at with the caveat of very low samples, mostly relying in one study for each relationship. Therefore, the results with respect to the antecedents to team viability should be interpreted with extreme caution.

Discussion

Organizations around the globe are relying on teams. This paper was a response to calls to investigate the conceptual and empirical links between team behaviors, attitudes, and cognitions as antecedents to team outcomes. Some of these have shown to be important across different outcomes, including leadership, conflict, collective efficacy, cohesion, and trust.

Similar to previous meta-analytic findings that task focused leadership and person-focused behaviors were related to team effectiveness and team productivity (Burke, Stagl, Klein, Goodwin, Salas, & Halpin, 2006), we found leadership to be positively related to performance. Furthermore, our results also show evidence for assertions such that when team members evaluate their team leaders positively, they are both more cohesive and satisfied (Gomes, Weber, Brown, & Tarba,2011; Wolfram & Mohr, 2009). Within leadership, leadership behaviors and leadership styles represent the most commonly examined aspects of leadership, leaving future research to scrutinize the relationship regarding shared or collective leadership (see Wang, Waldaman, & Zhang, 2014, for an exception).

Conflict, for instance, is often broken down into task and relationship conflict. Task conflict refers to disagreement regarding ways to approach the job, whereas relationship conflict is more related to interpersonal tension. While task conflict has been show to be less detrimental than relationship conflict (de Wit, Greer, & Jehn, 2012), both types of conflict showed a strong negative correlation to satisfaction (De Dreu & Weingart, 2003). Turning conflict into something positive has been a constant struggle within diverse teams literature (e.g., van Jaarsveldt & Joubert, *in press*). It has emerged as part of team development that influences team learning (Raes, Vanderhoven, & Schellens,2015), but also making team members more dissatisfied (Cox, 2003). Consequently, the emergence of conflict is likely to be detrimental to team learning for being associated with process loss. Our results suggest that conflict has a moderate negative relationship with team learning and team satisfaction. However, given that team learning findings only represented one study, we further break down the relationship between conflict and team satisfaction. Consistent with previous research, task and relationship conflict showed moderate negative relationships with team satisfaction.

The most influential inputs, however, were attitudinal emergent states when considering all outcomes, are likely to be due to the affective nature of team satisfaction and viability constructs. Collective efficacy (CE) and trust, for instance, were shown to play an important role to a number of team outcomes. CE is the idea that team member's beliefs of their team's capabilities converge and thus can be conceptualized at the team level (Kozlowski & Klein, 2000). Positive beliefs about members' capability to perform effectively leads to better performance (Gully, Incalcaterra, Joshi & Beaubien, 2002; Jung & Sosik, 2003), in both lab and field settings (Katz-Navon & Erez, 2005; Scott-Young & Samson, 2009; Stajkovic, Lee, Nyberg, 2009). This was also true in our meta-analysis, in addition to the relationship of CE and team viability. Even just the belief that the team can manage conflict can increase their viability (Jehn, Greer, Levine, & Szulanski, 2008). Considering the definition of viability to be the desire to remain a part of the same team in the future, this is highly related to whether the team members perceive their team as strong and with the potential for being successful. On a similar note, trust –or the willingness to be vulnerable to another party (see Mayer, Davis, & Schoorman,1995)– shows a strong positive relationship to team learning, explaining 42% of the variance in team learning. Trust has also been more often empirically linked to satisfaction within the team. Specifically, higher levels of trust translate to higher satisfaction (Costa, 2003; Gladstein, 1984). Within the team, trust can lead to satisfaction between dyads, or pairs of individuals who share interdependent goals (Smith & Barclay, 1997). Accordingly, our meta-analytic results support these claims.

Last but not least, team cohesion which has been investigated as an antecedent to team outcomes for over 60 years shows strong evidence based both within the military and broader literature. Several meta-analyses emerged to support the cohesion-performance link (e.g., Beal, Cohen, Burke, McLendon, 2003; Evans & Dion, 1991; Gully, Devine, & Whitney, 1995; Oliver, 1999). Our meta-analytic results support prior findings, team cohesion explains a significant 14% of the variance and is positively related to team performance, team learning, and team viability. While these findings for team viability are not based on a wealth of research, cohesion has shown to significantly increase team viability (Barrick, Stewart, Neubert, & Mount, 1998). Cohesion is considered one of the main antecedents of team effectiveness (Carron & Brawley, 2000), thus these positive outcomes of cohesion come without surprise.

So what do these findings mean in terms of GIFT and the updating of the original team state models proposed by Sottilare et al. (2011)? With respect to *team performance* we would update the affect state models to include trust, collective efficacy, cohesion, and psychological safety. Team communication state models would reflect communication, coordination, reflexivity, mutual support, conflict, leadership, and organizational citizenship behaviors. Team cognitive state models would include: team mental models, transactive memory systems, and situation awareness. With respect to *team satisfaction* the affect state models would be updated to include: collective efficacy, cohesion, trust, and psychological safety (with less confidence in the later two constructs). Communication state models would be reflected by communication, coordination, reflexivity, mutual support, conflict, leadership, and conflict management. Finally, team cognitive state models would be most likely reflected by team mental models and transactive memory systems; although the low k makes these cautionary. With respect to *team learning* as an outcome the team affect state models would include trust, conflict management, cohesion, and psychological safety (with less confidence in the later three due to low k). Team communication state models would include communication, coordination, reflexivity, conflict, and prior work would suggest leadership. There were not enough studies to update the team cognitive state models with respect to the outcome of team learning. Finally, with respect to *team viability* team state models may be updated in the following manner although due to low k all results were cautionary: (1) team affect: collective efficacy, cohesion, and psychological safety; (2) team communication: conflict, coordination; and (3) team cognitive state models: transactive memory systems.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

This research was an effort to capture team dynamics in a more integrative manner. Team effectiveness was conceptualized as consisting of the following four outcomes: team performance, team satisfaction, team viability, and team learning. These outcomes were chosen as they most often reflect team effectiveness within the conceptual models that appear in the literature. Team learning, while not always traditionally conceptualized as a team performance outcome, was included because within complex, mission critical environments, the team’s ability to learn and adapt is a hallmark of effectiveness.

This study is not without limitations. In a number of links, especially those with viability and learning, the amount of studies was very small. The main gap remains going beyond team performance as an outcome, and the relationship of cognition to other team variables. More work is needed to investigate what facilitates other team effectiveness outcomes beyond team performance. As future directions, we suggest taking a broader approach and refining the ontologies and team state models. More specifically, we suggest an approach that not only looks at the relationships in silos, but it can take into account the simultaneous importance of other team constructs. In essence, this would allow examination of the relative importance of each antecedent in predicting the outcome of interest. Due to the nature of the research some of the constructs will drop out due to a lack of studies examining the intercorrelations among variables, but looking at the relative importance along with what was reported here will provide further confidence in results. When constructs appear to have significant and direct relationships in both sets of analyses, more confidence is afforded.

Furthermore, in terms of the meaning and applicability of this work for GIFT, we encourage researchers to begin identifying behavioral markers for those constructs that explain the most variance in key outcomes. Specifically, while initial team models have been theorized, further work is needed to identify a complete design architecture, including behavioral markers and metrics that can be used to model team processes and performance in GIFT. Traditionally most of these constructs have been assessed via self-report and have not been translated into behavioral markers that can be captured via the training system. Development of such behavioral markers, especially with respect to attitudinal and cognitive variables represents a large move forward in creating an intelligent team tutor.

Future research should also take the dynamics of teams even more seriously by incorporating the component of time. Marks, Mathieu, and Zaccaro (2001) have suggested a taxonomy of action, transition, and interpersonal processes, suggesting that the temporal component should be brought to the forefront. This taxonomy is later supported by LePine’s (2008) meta-analysis. However, we could not take time into account due to the limited number of articles that dealt with such issue. This paper still allows for a more holistic picture of how team constructs influence team outcomes. The examination of moderators and inclusion of time can only strengthen our current findings via a more nuanced approach.

REFERENCES

Arthur, W., Jr., Bennett, W., Jr., & Huffcutt, A. I. (2001). Conducting meta-analysis using SAS. Mahwah, NJ: Erlbaum.

Barrick, M. R., Stewart, G. L., Neubert, M. J., & Mount, M. K. (1998). Relating member ability and personality to work-team processes and team effectiveness. *Journal of Applied Psychology*, 83(3), 377.

Beal, D. J., Cohen, R. R., Burke, M. J., & McLendon, C. L. (2003). Cohesion and performance in groups: A meta-analytic clarification of construct relations. *Journal of Applied Psychology*, 88(6), 989-1004.

Bell S. T.(2007). Deep-level composition variables as predictors of team performance: A meta-analysis. *Journal of Applied Psychology,* 92, 595-615.

Burke, C. S., Stagl, K. C., Klein, C., Goodwin, G.F., Salas, E., & Halpin, S.M. (2006). What type of leadership behaviors are functional in teams? A meta-analysis. *The Leadership Quarterly*, 17, 288-307.

Campion, M. A., Medsker, G. J., & Higgs, A. C. (1993). Relations between work group characteristics and effectiveness: Implications for designing effective work groups. *Personnel psychology*, 46(4), 823-847.

Cannon-Bowers, J. A. & Bowers, C. (2011). Team development and functioning. In S. Zedeck (Ed.) *APA handbook of industrial and organizational psychology, Vol 1: Building and developing the organization* (pp. 597-650). Washington, DC, US: American Psychological Association.

Carron, A. V., & Brawley, L. R. (2000). Cohesion conceptual and measurement issues. *Small Group Research,* 31(1), 89-106.

Costa, A. C. (2003). Work team trust and effectiveness. *Personnel Review*, 32(5), 605-622.

Cox, K. B. (2003). The effects of intrapersonal, intragroup, and intergroup conflict on team performance effectiveness and work satisfaction. *Nursing Administration Quarterly, 27*(2), 153-163.

De Dreu, C. K., W., & Weingart, L. R. (2003). Task versus relationship conflict, team performance, and team member satisfaction: A meta-analysis. *Journal of Applied Psychology*, 88(4), 741-749.

de Wit, F. R., Greer, L. L., & Jehn, K. A. (2012). The paradox of intragroup conflict: a meta-analysis. *Journal of Applied Psychology*, 97(2), 360.

Dyer, L. (1984). Studying human resource strategy: An approach and an agenda. *Industrial Relations: A Journal of Economy and Society,* 23(2), 156-169.

Evans, C. R., & Dion, K. L. (1991). Group cohesion and performance a meta-analysis. *Small group research*, 22(2), 175-186.

Gladstein, D. L. (1984). Groups in context: A model of task group effectiveness. *Administrative Science Quarterly*, 499-517.

Gomes, E., Weber, Y., Brown, C., & Tarba, S. Y. (2011). Mergers, acquisitions, & strategic alliances. Basingstoke, UK: Palgrave Macmillan.

Gully, S. M., Devine, D. J., & Whitney, D. J. (1995). A meta-analysis of cohesion and performance effects of level of analysis and task interdependence. *Small Group Research*, *26*(4), 497-520.

Gully, S. M., Incalcaterra, K. A., Joshi, A., & Beaubien, J. M. (2002). A meta-analysis of team efficacy, potency, and performance: Interdependence and level of analysis as moderators of observed relationships. *Journal of Applied Psychology*, 87, 819–832.

Hackman, J. R. (1987). The design of work teams. In J. W. Lorsch (Ed.) Handbook of organizational behavior (pp. 315-342). Englewood Cliffs, NJ: Prentice-Hall.

Hunter, J. E. and Schmidt, F. L. (2004). Methods of meta-analysis. Newbury.

Jehn, K.A., Greer, L.L., Levine, S., & Szulanski, G. (2008). The effects of conflict types, dimensions, and emergent states on group outcomes. *Group Decision and Negotiation*, 17, 465–495.

Jung, D. & Sosik, J. (2003). Group potency and collective efficacy: Examining their predictive validity, level of analysis, and effects of performance feedback on future group performance. *Group and Organization Management*, 28, 366–391.

Katz-Navon, T. and Erez, M. (2005). When Collective and self-efficacy affect team performance, the role of task interdependence. Small Group Research, 36, 437-465.

Klein, C., DiazGranados, D., Salas, E., Le, H., Burke, C. S., Lyons, R., & Goodwin, G. F. (2009). Does team building work? Small Group Research.

Kozlowski S. W. J. & Klein K. J. (2000). A multi-level approach to theory and research in organizations: Contextual, temporal, and emergent processes. In Klein K. J. and Kozlowski S. W. J. (Eds.) *Multilevel theory, research, and methods in organizations* (pp. 3-90). San Francisco: Jossey-Bass.

LePine, J. A., Piccolo, R. F., Jackson, C. L., Mathieu, J. E., & Saul, J. R. (2008). A meta-analysis of teamwork processes: tests of a multidimensional model and relationships with team effectiveness criteria. *Personnel Psychology*, 61(2), 273-307.

Marks, M. A., Mathieu, J. E., & Zaccaro, S. J. (2001). A temporally based framework and taxonomy of team processes. *Academy of Management Review*, 26, 356–376.

Mathieu, J. E., Marks, M. A., & Zaccaro, S. J. (2001). Multi-team systems. In N. Anderson, D. Ones, H. K. Sinangil, & C. Viswesvaran (Eds.) *International handbook of work and organizational psychology* (pp. 289–313). London: Sage.

Mayer, R. C., Davis, J. H., & Schoorman, F. D. (1995). An integrative model of organizational trust. *Academy of Management Review*, 20, 709-734.

Mesmer-Magnus, J. & Viswesvaran, C. (2010). The role of pre-training interventions in learning: A meta-analysis and integrative review. *Human Resource Management Review*, 20(4), 261-282.

Nunnally, J. (1978). Psychometric theory. New York, NY: McGraw-Hill.

Oliver, R.L. (1999). Whence consumer loyalty? *Journal of Marketing*, 63, 33-44.

Raes, A., Vanderhoven, E., & Schellens, T. (2015). Increasing anonymity in peer assessment by using classroom response technology within face-to-face higher education. *Studies in Higher Education*, 40(1), 178-193.

Salas, E., DiazGranados, D., Klein, C., Burke, C. S., Stagl, K. C., Goodwin, G. F., & Halpin, S. M. (2008). Does team training improve team performance? A meta-analysis. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 50(6), 903-933.

Salas, E., Dickinson, T. L., Converse, S. A., & Tannenbaum, S. I. (1992). Toward an understanding of team performance and training. In R. W. Swezey, E. Salas, R. W. Swezey, E. Salas (Eds.) , *Teams: Their training and performance* (pp. 3-29). Westport, CT, US: Ablex Publishing.

Scott-Young, C. & Samson, D. (2009). Team management for fast projects: an empirical study of process indutries. *International Journal of Operations & Production Management*, 29(6), 612-635.

Smith, J. B., & Barclay, D. W. (1997). The effects of organizational differences and trust on the effectiveness of selling partner relationships. *Journal of Marketing*, 3-21.

Smith-Jentsch, K. A., Cannon-Bowers, J. A., Tannenbaum, S. Il, & Salas, E. (2008). Guided team self-correction: Impacts on team mental models, processes, and effectiveness. Small Group Research, 39(3), 303-327.

Smith-Jentsch, K. A., Zeisig, R. L., Acton, B., & McPherson, J. A. (1998). Team dimensional training: A strategy for guided team self-correction. In J. A. Cannon-Bowers and E. Salas (Eds.) *Making decisions under stress: Implications for individual and team training* (pp. 271-297). Washington, DC, US: American Psychological Association.

Sottilare, R. A., Holden, H. K., Brawner, K. W., & Goldenberg, B. S. (2011). Challenges and emerging concepts in the development of adaptive, computer-based tutoring systems for team training. Army Research Lab Orlando FL Human Research and Engineering Directorate.

Stajkovic, A., Lee, S., Nyberg, A. (2009). Collective efficacy, group potency, and group performance: Meta-analyses of their relationships, and test of mediation model. *Journal of Applied Psychology*, 94, 814-828.

van Jaarsveldt, D. E., & Joubert, A. (*in press*). Navigating Diversity With Nursing Students Through Difficult Dialogues: A Qualitative Study. *International Journal of Africa Nursing Sciences*.

Wang, D., Waldman, D. A., & Zhang, Z. (2014). A meta-analysis of shared leadership and team effectiveness. *Journal of Applied Psychology*, 99(2), 181.

Wolfram, H. J. and Mohr, G. (2009). Transformational leadership, team goal fulfillment, and follower work satisfaction: The moderating effects of deep-level similarity in leadership dyads. *Journal of Leadership and Organizational Studies*, 15, 260–274.

ABOUT THE AUTHORS

Dr. C. Shawn Burke *is an Associate Professor (Research) at the Institute for Simulation and Training of the University of Central Florida. Her expertise includes teams and their leadership, team adaptability, team training, measurement, evaluation, and team effectiveness. Dr. Burke has published over 80 journal articles and book chapters related to the above topics and has presented/had work accepted at over 170 peer-reviewed conferences. She is currently investigating issues surrounding: leadership within virtually, distributed teams, team cohesion, issues related to multi-cultural team performance and multiteam systems. The above work is conducted with an interest in team leadership and teams operating in complex environments.*

Jennifer Feitosa *is a doctoral candidate in the Industrial/Organizational Psychology program at the University of Central Florida, where she also earned a M.S. degree in 2013. As a graduate research associate at the Institute for Simulation and Training, her research interests include teams and culture, with an overarching emphasis on statistical methods. She is currently involved in projects investigating aspects of team dynamics, funded by the National Aeronautics and Space Administration and Army Research Laboratory. She has co-authored nine publications and has personally presented 23 papers or posters at professional conferences.*

Dr. Eduardo Salas *is Trustee Chair and Pegasus Professor of Psychology at the University of Central Florida (UCF). He also holds an appointment as Program Director for Human Systems Integration Research Department at UCF's Institute for Simulation & Training. Previously, he was a Senior Research Psychologist and Head of the Training Technology Development Branch of the Naval Air Warfare Center-Orlando for 15 years. During this period, Dr. Salas served as a principal investigator for numerous R&D programs focusing on teamwork, team training, simulation-based training, decision-making under stress, learning methodologies and performance assessment.*

THEME III: INSTRUCTIONAL MANAGEMENT

Unsupervised Discovery of Tutorial Dialogue Modes in Human-to-Human Tutorial Data

Vasile Rus[1], Nabin Maharjan[1], and Rajendra Banjade[1]
[1]Department of Computer Science and Institute for Intelligent Systems – The University of Memphis

INTRODUCTION

A key part of the Generalized Intelligent Framework for Tutoring (GIFT; Sottilare, Brawner, Goldberg, & Holden, 2012) is the instructional management component whose role is to "integrate pedagogical best practices." To this end, we present in this paper our exploration of best practices by professional tutors in a commercial tutoring service context. In particular, we analyze human-to-human tutoring sessions provided to us by Tutor.com, a leading provider of human tutoring services. The main form of interaction in these tutorial sessions is chat-based conversation and therefore our focus is on analyzing dialogue-based tutoring sessions. Our plan for exploring professional tutors' best practices is to explore, analyze, and characterize patterns of actions (dialogue moves) in tutorial sessions conducted by these tutors.

We adopt a data-driven approach to discovering best practices in the form of patterns of tutor and tutees' actions. Indeed, the approach we follow starts with segmenting the dialogue-based interactions between tutors and tutees into sequences of dialogue acts based on the language-as-action theory (Austin, 1962) and then automatically infer patterns over these sequences of dialogue acts in the form of hidden states using an unsupervised, data-driven approach. We view the discovered patterns as the result of deliberate conversational and pedagogical strategies by the tutor and tutee and therefore offer an interpretation of the hidden states accordingly, by relating the hidden states to conversational and pedagogical goals following a methodology inspired from Cade, Copeland, Person, and D'Mello (2008), Boyer Phillips, Ingram, Ha, Wallis, Vouk, and Lester (2011), and Jeong, Gupta, Roscoe, Wagster, Biswas, and Schwartz, (2008). Patterns of actions in tutorial dialogues can be associated with general conversational segments (e.g., openings/closings) and task-related and pedagogical goals (e.g., scaffolding). Such patterns are called dialogue modes in the literature (Cade et al., 2008) and, when they can be linked to pedagogical goals, are regarded as tutorial strategies (Morrison & Rus, 2014).

An interesting aspect of research in this area of discovering meaningful patterns in tutoring, i.e. the underlying instructional and learning strategies, is what the focus of analysis should be: what tutors do (which is about discovering instructional or, in our case, tutorial strategies), what students do (which is about discovering learning strategies), or characterizing the nature of the tutor-learner interaction which is about discovering the structure, patterns, and dynamics of the interaction. Previous research studies addressed each of the above foci: tutor actions (Jackson, Person, & Graesser, 2004), tutee actions (Jeong et al., 2008), and both tutor-tutees actions (Litman & Forbes-Riley, 2006; Ohlsson, DiEugenio, Chow, Fossati, Lu, & Kershaw, 2007; Boyer et al., 2011). We focus here primarily on studying the nature of the tutor-learner interaction as the better alternative for the time being as is explained later. Interestingly, we discover patterns and corresponding latent processes across tutors, students, and topics. Access to a large pool of tutorial sessions (245,064 sessions) from many tutors and students across two domains (Algebra and Physics) made this study possible.

It should be noted that focusing on discovering professional tutors' or learners' strategies from previously collected data could be challenging for several reasons. First, tutors' strategies may or may not trigger effective learner strategies which make it hard to focus on either tutors' strategies or learners' strategies without a careful controlled experiment or careful selection of instances data. Furthermore, we do not focus particularly on what tutors do because the jury is still out with respect to whether this is the best way to move forward given the available data. In general, expert tutoring is believed to lead to more effective learning. An intriguing result reported recently indicates that the expertise level of tutors, as measured by their experience or how much they are paid, does not impact average learning gains nor did the tutor experience explain a significant portion of the variance in learning gains (Ohlsson et al., 2007). Indeed, this might suggest that focusing solely on what tutors do might not be the best way moving forward. It should be noted that Ohlsson and colleagues used a very small number of tutors in their study.

Based on our data analyses we learned that there is no significant correlation (0.03; p=0.260, N=1,038) between tutoring experience, as measured by months of tutoring, and successful sessions. However, in our case we only had indirect measures of session success in the form of post-hoc human expert judgments of student learning as opposed to more standard measure of learning, i.e. using a pre- and post-test. It should be noted that we did find a significant albeit modest correlation (0.07; p=0.033, N=891) between human expert (tutor mentors) judgments of tutor expertise and human tutor expert judgments of student learning of tutorial sessions. Indeed, tutoring expertise and its relation to learning is yet to be elucidated. It is beyond the scope of this paper to address the topic of tutoring expertise. Given that our data was collected from professional tutors, i.e. tutors that tutor to make a living, the results will be interpreted with this qualification in mind. That is, we discover patterns in professional tutor sessions.

As already indicated, our approach is data-driven, making no assumptions about the number or nature of the underlying tutorial interaction patterns, which we infer in the form of hidden or latent states with the help of the Hidden Markov Model (HMM) framework. The alternative is to have a set of predefined modes proposed by experts and then simply develop automated methods to label tutorial dialogues with the predefined set of modes, which is the framework adopted by Rus, Niraula, Maharjan, and Banjade (2015). Rus and colleagues developed a method to label tutorial dialogues with a set of expert-defined dialogue modes using a supervised method. That is, they learned from human-annotated data the signatures of various dialogue modes using a sequence labeling framework, i.e. Conditional Random Fields (CRFs; Lafferty, McCallum, Pereira, 2001). In contrast, we use a totally unsupervised method. We ask two fundamental questions: how many dialogue modes are intrinsically in the data and what is the nature of these data-suggested dialogues modes. Our work complements theoretically-driven efforts to define dialogue modes. We do relate our discovered modes to expert-predefined modes in order to identify similarities, which could be a validation of the expert-defined modes, and dissimilarities, which would enable the discovery of modes that were not previously identified by experts.

The rest of the paper is organized as following. Next, we briefly overview relevant previous work. We then detail our approach followed by a section dedicated to the results of our experiments. We wrap-up with a set of conclusions and recommendations for GIFT.

PREVIOUS WORK

Discovering the structure of tutorial dialogues has been a main goal of the intelligent tutoring research community for quite some time. For instance, Graesser, Person, and Magliano (1995) explored collaborative dialogue patterns in tutorial interactions and proposed a five-step general structure of collaborative problem solving during tutoring.

Over the last decade, the problem of automated discovering of the structure of tutorial dialogues has been better formalized and also investigated more systematically using more rigorous analysis methods (Cade, et al., 2008; Jeong, et al., 2008; Chi, VanLehn, & Litman, 2010; Boyer, et al., 2011). For example, tutoring sessions are segmented into individual tutor and tutee actions and statistical analysis and artificial intelligence methods are used to infer patterns over the tutor-tutees action sequences. The patterns are interpreted as tutorial strategies or tactics which can offer both insights into what tutors and students do and guidance on how to develop more effective intelligent tutors that implement these strategies automatically, which is very relevant to GIFT. An interesting line of research in this area is about the discovery of dialogue modes, which is reviewed next.

Dialogue Mode Identification

Dialogue modes are sequences of dialogue acts that correspond to general conversational segments of a dialogue, e.g. an *Opening* mode corresponds to the first phase of the dialogue when the conversational partners greet each other, or to segments associated with pedagogical goals, e.g. a *Scaffolding* mode would correspond to the tutorial dialogue segment when the learner works on something and the tutor scaffolds the learner activity. Discovering such dialogue modes could be extremely useful for understanding what exactly professional tutors do during a tutoring session and transfer that understanding to the development of Intelligent Tutoring Systems (ITSs).

Previous research on dialogue modes relied on both analytical and automated approaches. Cade and colleagues (2008) defined, based on a manual analysis, a set of eight mutually exclusive tutorial modes: *introduction*, *lecture*, *highlighting*, *modeling*, *scaffolding*, *fading*, *off-topic*, and *conclusion*. An interesting aspect of their analysis is the granularity at which they define pedagogically important modes such as scaffolding, modeling, and fading. In their approach, the modes correspond to either the tutor or the students or both focusing on solving a full problem. In our approach, we used a different definition of modes proposed by Morrison, Nye, Samei, Datla, Kelly, and Rus (2014). In this approach, tutor and student could switch between proposed modes while working on a particular problem. That is, a particular mode is not associated with one problem solving task but rather with parts of such a problem solving task.

Boyer and colleagues (2011) used sequences of task actions and dialogue acts to automatically discover signature action sequences based on HMM model fitting. Furthermore, they related the automatically discovered modes to student learning. They discovered anywhere between 8 to 10 hidden states, or modes, depending on the tutor. Their set of modes include: *Correct Student Work*, *Tutor Explanations with Feedback*, *Tutor Explanations with Assessing Questions*, *Student Work with Tutor Positive Feedback*, *and Student Acting on Tutor Help*. Importantly, their discovery process is applied to sessions of individual tutors and not across tutors, as is the case in our work.

Jeong and colleagues (2008) used HMMs to discover learning strategies in a learning-by-teaching environment. More specifically, they used HMMs to discover patterns in students' actions in response to meta-cognitive prompting in such learning-by-teaching environments. Jeong and colleagues used six main activities as observables to infer the hidden states, i.e. the learning strategies. It should be noted that because the six activities are actions taken by the learners, the discovered hidden states are indeed learning strategies as opposed to instructional or tutoring strategies, which would be characterized by actions taken by the instructor or tutor. They used the dominant activity or pair of activities in each hidden state to label the discovered states; they dropped any activity that occurred less than 10%. Interestingly, they interpreted both the hidden states and the transitions among these hidden states as learning strategies. That is, once they discovered the major hidden states and the dominant activities in these states, they looked at well-traveled paths of hidden states (i.e. sequences of hidden states with high transition probabilities) and interpret them as learning strategies that make sense from a meta-cognitive point of view. The learner strategies are then viewed as aggregate states and used to build an aggregate-state transition resulting in a second degree state-transition automaton in which the states are associated with the previously discovered learner strategies. Their set of hidden states and aggregate-states is small too (<10).

THE APPROACH

Our approach to automatically discover dialogue modes from human-to-human tutorial dialogues is to map each dialogue into sequences of dialogue acts based on the language-as-action theory (Austin, 1962; Searle, 1969) and then infer patterns over these sequences of dialogue acts using the unsupervised Hidden Markov Models framework (Rabiner, 1989). The inferred patterns in the form of latent states are then interpreted. The details of this approach are presented next.

From Utterances to Dialogue Act Sequences

We regard tutors' and students' utterances in a tutorial dialogue as encoding actions based on the language-as-action theory according to which when people say something they do something (Austin, 1962; Searle, 1969). A speech or dialogue act is a construct in linguistics and the philosophy of language that refers to the way natural language performs actions in human-to-human language interactions, such as dialogues. Its contemporary use goes back to John L. Austin's theory of locutionary, illocutionary and perlocutionary acts (Austin, 1962).

The notion of speech act is closely linked to the illocutionary level of language, which consists of the actual utterance and its exterior meaning. The idea of an illocutionary act, e.g. asking questions ("*Is it snowing?*") or giving a warning ("*The floor is wet!*"), can be best captured by emphasizing that "*by saying something, we do something*" (Austin, 1962). The illocutionary force is not always obvious and could consist of different components. As an example, the phrase "*It's cold in this room!*" might be interpreted as having the intention of simply describing the room, or criticizing someone for not keeping the room warm, or requesting someone to close the window, or a combination of the above. A speech act could be described as the sum of the illocutionary forces carried by an utterance. It is worth mentioning that within one utterance, speech acts can be hierarchical, hence the existence of a division between direct and indirect speech acts, the latter being those by which one says more than what is literally said, in other words, the deeper level of intentional meaning. In the phrase, "*Would you mind passing me the salt?*", the direct speech act is the request best described by "*Are you willing to do that for me?*" while the indirect speech act is the request "*I need you to give me the salt.*" In a similar way, in the phrase "*Bill and Wendy lost a lot of weight with a diet and daily exercise.*" the direct speech act is the actual statement of what happened "*They did this by doing that.*", while the indirect speech act could be the encouraging "*If you do the same, you could lose a lot of weight too.*"

The present work assumes there is one direct speech act per utterance. This simplifying assumption was appropriate for automating the speech act (or dialogue act) discovery process. We did differentiate between top-level dialogue acts and second-level subacts but this was just a hierarchical organization that allowed us to analyze and process the dialogues at different levels of abstractness. A combination of an act and subact uniquely identified, in our work, the direct speech act associated with an utterance. In fact, in this study we used just the set of dialogue acts (no subacts) due to various reasons explained later.

In our case, to map utterances in a tutorial dialogue into corresponding dialogue acts or actions we use a predefined dialogue or speech act taxonomy. The taxonomy was defined by educational experts and resulted in a two-level hierarchy of 15 top-level dialogue acts and a number of dialogue subacts. The exact number of subacts differs from dialogue act to dialogue act. The overall, two-level taxonomy consists of 126 unique dialogue-act+subact combinations (Morrison, et al., 2014). It should be noted that automatically discovered dialogue act taxonomies are currently being built (Rus, Graesser, Moldovan, & Niraula, 2012) but it is beyond the scope of this paper to automatically discover the dialogue acts in our tutoring sessions. An example of mapping dialogues into dialogue acts sequences is shown in Table 1 where the column titled *Dialogue Act* contains the dialogue act labels corresponding to each of the utterances in the column *Utterance*. It should be noted that a dialogue turn, i.e. the contiguous span of the dialogue during which a speaker has the floor, may contain multiple utterances from the same speaker as can be seen in Table 1.

Table 1. A snapshot of a tutorial session and the corresponding expert-labeled dialogue acts for each utterance.

Speaker	Utterance	Dialogue Act
T(utor)	*Note that the small angle approximation gives us y = lambda L / 2 w*	Directive
S(tudent)	*i think i did that*	Assertion
T	*Eliminating the need for sin*	Directive
T	*Okay*	Expressive
T	*Oh, and forget the 2 in the denominator.*	Directive
T	*y = lambda L / w*	Assertion
T	*Where y is the width of the beam*	Assertion
S	*ok well i got an angle of.001*	Assertion
S	*and multiplied by 2*	Assertion
S	*then i did tan of that angle times L*	Assertion
S	*Is that the same thing*	Request
S	*i multiplied that final answer i got for y by 2*	Assertion
T	*That's what you need to do*	Assertion

Dialogue Mode Discovery Using Hidden Markov Models

Once a tutorial dialogue was mapped into a sequence of dialogue acts, our goal was to automatically infer patterns over sequences of dialogue acts in the form of hidden states using an unsupervised, data-driven approach based on the Hidden Markov Models framework (HMMs; Rabiner, 1989). We applied HMMs, as in Boyer and colleagues (2011) and Jeong and colleagues (2008), to mine chunks of tutorial dialogue actions that can be associated with general conversational segments (e.g., openings/closings) and task-related and pedagogical goals (e.g., scaffolding).

Markov models are a statistical theory used to model stochastic sequential processes. In particular, Markov models are used to investigate and model processes that satisfy the Markov assumption according to which the current state of the process is dependent only on a limited number of previous states, i.e. the current state is only dependent on the recent history of events. Typically, the current state is assumed to depend only on the previous state in which case we have a first order Markov model.

When the states of the stochastic process are not directly observable, i.e. they are hidden, we are dealing with Hidden Markov Models (HMMs; Rabiner, 1989). In such cases, we only learn about the hidden states indirectly through a set of observations whose relationship to the hidden states is modeled through another set of stochastic processes described by the so-called emission probability distributions. In a way, a HMM can be regarded as two layers of stochastic processes: a stochastic process among the hidden states (hidden layer) and a set of stochastic processes for each of the hidden states and the observations (partially observable layer). Because the hidden states are not directly observable and are therefore discovered, the HMMs are suitable for understanding data in terms of latent processes, i.e. processes that might have generated the observed data. In our case, we assume that there is an underlying process that governs the tutor-tutee interaction and the states of this process correspond to dialogue modes.

Furthermore, HMMs have computational advantages such as the availability of unsupervised algorithms that can learn the parameters of the model and to some degree the structure of the model from raw observations. Before we detail the learning and model selection process, we describe formally the HMMs.

An HMM is a quintuple (S, O, A, B, Π) consisting of:

- a set of states S ($\{s_1, s_{2,} s_{3,\dots,} s_N\}$), which are not observed directly;
- a set of observations O ($\{o_1, o_{2,} o_{3,\dots,} o_M\}$) or alphabet;
- a set of transition probabilities a_{ij} specifying the probability of making a transition from state s_i to state s_j: A=$\{a_{ij}\}$;
- a set of emission probabilities $b_j(k)$ specifying the probability of state s_j emitting observation o_k: B=$\{b_j(k)\}$;
- a set of initial state probabilities $\Pi = \{\pi_i\}$ specifying the likelihood of a particular state s_i being the initial state of the stochastic process.

As mentioned before, there exists a learning procedure that can infer the parameters of a HMM given a set of observations (dialogue act sequences in our case). The basic idea is to fit a model that best explains the data as measured by the likelihood of the data (actually, for computational reasons log-likelihood is used). The parameter learning algorithm is based on the Expectation-Maximization (E-M) paradigm. It starts with a random initial assignment of parameters and following a hill-climbing process the parameters are adjusted until convergence occurs, i.e. no more improvements in the likelihood are detected, or a threshold of iterations has been reached. The algorithm is unsupervised, which is a major advantage. The trade-off is that it might converge to a local minimum; also, it requires that the number of states, i.e. the structure, is provided as input. To address the former limitation, one can re-run the process many times with different parameter initializations, and then choose the model with the highest likelihood among the different runs.

The second issue of selecting the number of states requires a more elaborated discussion due to the complexity of the task. Indeed, an important aspect of HMM-based discovery processes is solving the model size problem, i.e. the number of latent or hidden states which characterizes, to some extent, the structure of the inferred HMM. HMMs of different sizes model data at different levels of abstraction (Li & Biswas, 2002). HMMs with a larger number of hidden states typically provide a more detailed description for data than models with fewer hidden states. On the other hand, as the size of the HMM grows the complexity of the model increases which often leads to an overfitting problem, i.e. the model captures the training data very well (with high accuracy) but does poorly on new, unseen data (low generalization). In addition, a model that is too complex makes the model interpretation task more difficult. As in many other machine learning algorithms, the challenge is to find the right balance between accuracy and generalization which is typically achieved by regularization, as explained next.

Using the traditional log-likelihood criterion is problematic as log-likelihood increases with the model complexity, i.e. it favors HMMs with larger number of states. This criterion was used by Boyer and colleagues (2011), for instance, to fit the best HMM to their tutorial data to discover dialogue modes.

To counter this tendency of log-likelihood to select more complex models, which overfit the data and generalize poorly, we employed a regularized criterion in which a penalty parameter that increases with the complexity of the underlying model is added to the log-likelihood term. The Bayesian Information Criterion (BIC) shown below is the regularized criterion we used. BIC represents an approximation of the posterior likelihood $P\ (H|E)$ (Schwarz, 1978; Heckerman, 1996; Li & Biswas, 2002).

$$BIC \approx \log P\left(E\middle|\hat{\lambda}, H\right) - \frac{d}{2}\log N$$

The BIC criterion contains two terms: the first term, $P\left(E\middle|\hat{\lambda}, H\right)$, is the likelihood of the data under the max likelihood parameter configuration obtained with the E-M algorithm, while the second term is a penalty term that increases with complexity of the model d. The model complexity of HMMs is O(n^2) where *n* is the number of states. N in the equation above is the number of training examples.

The best model selected using the Bayesian model selection criterion is the one that is compact in size and adequately describes the data. In other words, we follow Occam's razor principle according to which we should select the simplest model that best describes the data. It can be shown that BIC is equivalent to the Minimum Description Length principle (Mitchell, 1997).

Once the best HMM was selected, the resulting hidden states are interpreted based on the distribution of dialogue acts associated with these states as in Jeong and colleagues (2008) and Boyer and colleagues (2011). We will present the interpretation of these latent states in the *Results* section, which is next.

RESULTS

We conducted experiments with dialogue mode discovery based on HMMs on two data sets: (1) a very large sample of 245,064 sessions, which were automatically tagged with dialogue acts using an automated dialogue act classifier, and (2) a manually annotated data set of 1,438 sessions. The manually annotated data set consisted of 1,438 sessions which included 95,526 utterances generated by both tutors and students. This data set is a representative sample of the big dataset of 245,064 sessions. Sessions which lasted less than 5 minutes were eliminated from our analysis, as many of them are not true tutoring sessions, e.g. the session ends suddenly due to some networking issue. Also, due to limitations of the HMM software we used (JavaHMM), sessions longer than 210 dialogue acts are not properly processed, i.e. an underflow exception is triggered because of multiplying many very small numbers (probabilities values) for very long dialogues. The last row in Table 2 shows the actual number of sessions we used for dialogue mode discovery.

Table 2. Overview of the two datasets (#sessions): human tagged data and full data.

	Human Tagged	250k sessions
Original	1,438	245,064
>5min	1,306	222,369
HMM run	1,295	217,998

Dialogue Mode Discovery Using Hidden Markov Models

We focus on a purely discovery process in which the goal is to learn the optimal number of modes supported by the data. We are also interested in the nature of these data-supported modes, which we analyze in the next section when interpreting the discovered modes. We characterize the nature of the discovered modes in terms of dialogue act distributions by the tutor and by the tutee.

The discovery process relies on Hidden Markov Models (HMMs), described earlier, in which we repeatedly vary the number of hidden states assumed to be in the data. For each such number of states we evaluate the quality of the model inferred using the Bayesian Information Criterion (BIC) and, for comparison purposes, the log-likelihood. For a particular number of states, each model was trained using 50 different initializations to avoid local minima. For each number of states and each initialization of the model parameters, these models were trained for 50 iterations or until convergence, whichever came first, resulting in about 2500 training iterations for each number of states for a total of 29 x 2500 = 72,500 training iterations for one setup, e.g. using just acts or acts+subacts with speaker information, i.e. a tutor dialogue act versus a student dialogue act.

Results on Automatically Tagged Data

The chart in Figure 1 shows BIC values for various HMMs when varying the number of hidden states from 2 to 30 and using dialogue acts differentiated by speakers to infer the hidden states from the large dataset. We also added the log-likelihood in the chart– it is not visible because it overlaps the BIC curve entirely. Furthermore, we note that the values keep going up for both log-likelihood and the BIC. While this is not surprising for log-likelihood, it is a bit surprising for BIC which should penalize complex/larger models more, i.e. as the number of hidden states increases. This unusual behavior for BIC is due to the fact that the data is so large relative to the model sizes that we explored (n=2-30 states) that the likelihood term in BIC dominates the penalty term. There are several ways to address this issue: (1) give more weight to the penalty term for small and medium size models, or (2) increase the size of the models significantly, e.g. to thousands of hidden states, which would automatically increase the magnitude of the penalty term to values comparable to the max likelihood term for such large data. Either solution has interesting implications. The former would lead to an operational BIC at reasonable size models which, as illustrated in the next section, facilitates interpretation of the discovered states. On the other hand, giving more weight to the penalty term seems like an artificial step.

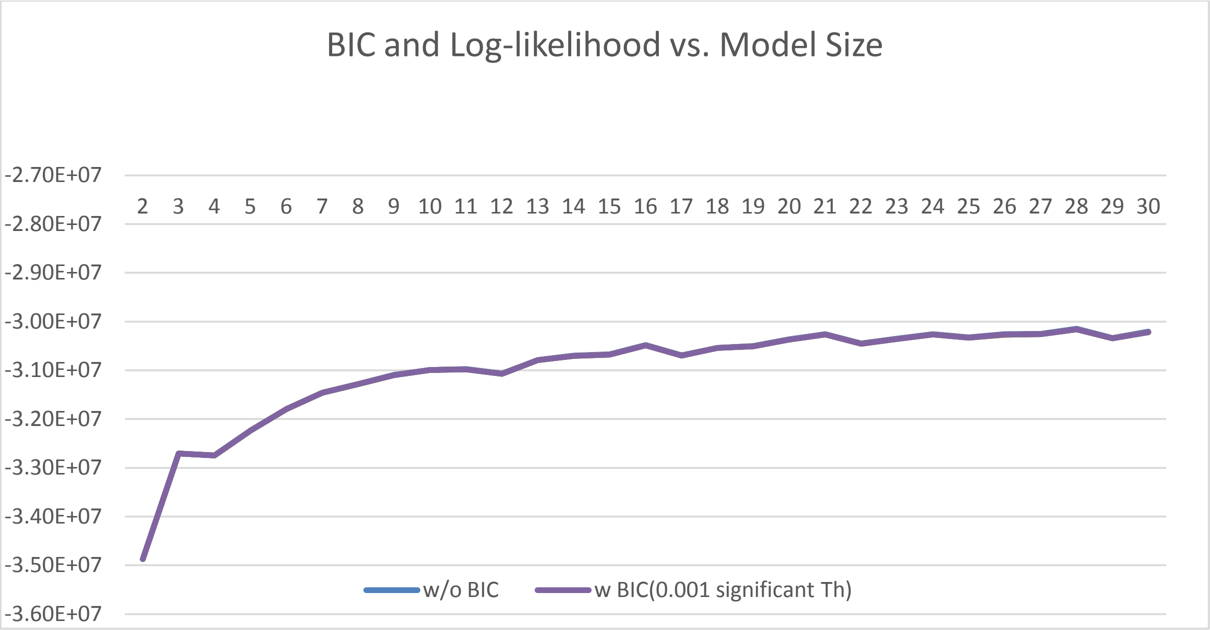


Figure 1. Trends for BIC and Log-likelihood model selection criteria as a function of model size (n=2-30) on the big dataset.

The latter solution of boosting the model size significantly would lead to models which are harder to interpret. However, it could be an interesting experiment as it might be the case that given the complexity of the task having thousands of discovered modes would better match a complex reality involving thousands of tutors and students and different topics. An alternative conclusion is that the data is so complex, any future analysis should focus on various, interesting subsets, e.g. just Physics sessions or students with a particular background. We will explore these alternative solutions in the future. Next, we focus on discovering the latent dialogue states obtained from the smaller, human tagged data, which, as we will see, leads to a clear optimum number of states.

It should be noted that we also experimented with discovering hidden states based on dialogue acts and subacts as well as differentiating such acts and subacts between speakers (tutor vs. student). However, our trials with several software packages for inferring HMMs failed to produce any useful output due to a too rich observation space, e.g. underflow exceptions occurred frequently.

Results on Human Tagged Data

Figure 2 shows the BIC and log likelihood ("without using BIC") for various HMMs when varying the number of hidden states from 2 to 30 on the annotated data set. The chart was obtained using dialogue acts, differentiated by tutor and student, as observations. When computing BIC, we ignored extremely small parameters in the model (<.001) as in Li and Biswas, (2002). The BIC line peaks at n=19 states which indicates that this is the optimum number of states supported by the data; this is the model that generalizes well without overfitting the data. As we notice from the chart, the log-likelihood keeps increasing with the model complexity as measured by the number of states on the x axis. Next, we focus on interpreting the 19 states of this best-fit HMM.

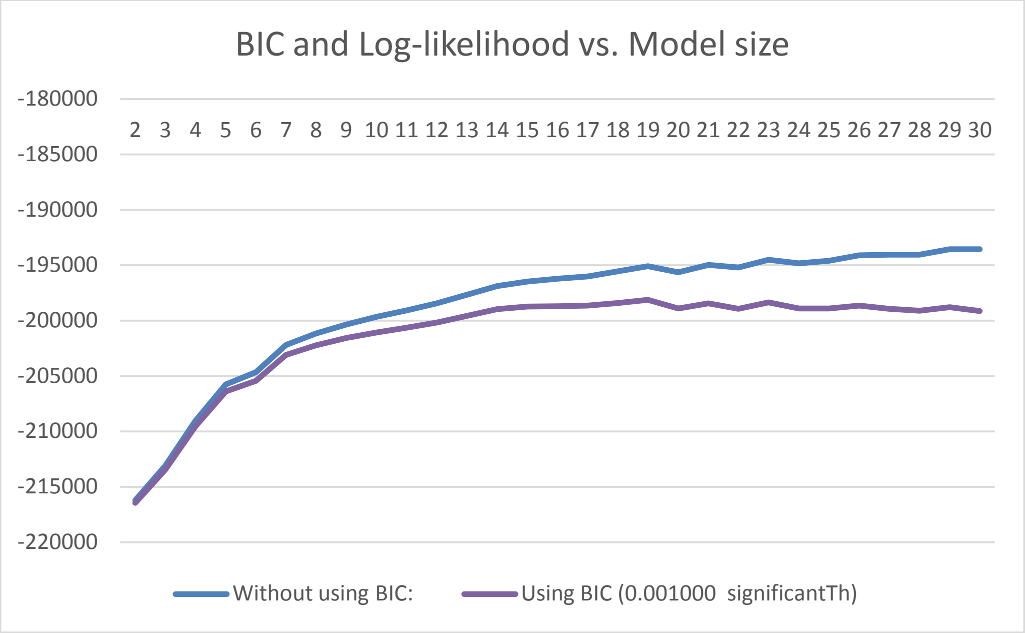


Figure 2. Trends for BIC and Log-likelihood criteria as a function of model size (n=2-30) on the human annotated dataset.

Interpretation of Results

The interpretation of the hidden states was inspired by previous work on the topic by Jeong and colleagues (2008) and Boyer and colleagues (2011). We used the following three methods to guide the interpretation process: (1) analyze the dialogue act mixtures for each of the hidden states as reflected in the emission probabilities of the HMM; (2) analyze dialogue act mixtures differentiated by speaker (tutor vs. student); (3) label sessions with the hidden states and manually inspect fragments of sessions for each of the hidden states.

We only retain for each hidden state the dominant moves/acts by the tutor and student (>5%) in order to interpret and label the discovered modes. Table 3 shows the interpretation for a 19-state HMMs based on the three dominant dialogue acts and their distribution as derived from the full dataset. That is, we ran the 19-state HMM on the full data set to assign states to each utterance and then derived for each hidden state the distribution of dialogue acts. For instance, state S1 in Table 3 is dominated by tutor’s explanations, expressives, and assertions. By comparison, we show in Table 4 the set of expert-defined modes and the corresponding dominant acts extracted from our human annotated dataset. Details of the 16-expert defined modes can be found in Morrison, Nye, Samei, Datla, Kelly, and Rus (2014). We briefly describe them below.

- *Opening*: The opening mode of the session (e.g. greetings, asking for help).

- *Problem Identification*: a mode in which the tutor seeks to “identify and achieve a clear understanding of the student’s needs/expectations.”

- *Assessment*: a mode in which the tutor “assesses the student’s level of understanding” in order to determine a “suitable starting point.”

- *Method Identification*: A mode during which the tutor attempts to determine if there is a particular method the student needs to use to solve the problem.

- *Method RoadMap*: A mode during which the tutor lays out the “game plan” for solving the problem.

- *Telling*: a mode in which the tutor asserts and explains, with relatively few student contributions mostly in the form of confirmations of understanding.

- *Modeling*: a mode in which the tutor is showing the student how to solve a problem, such as by completing one or more of the steps herself.

- *Scaffolding*: a mode in which the student is doing most of the work and the tutor making frequent contributions of assistance.

- *Fading*: a mode in which the student is working on the problem successfully, with only occasional contributions from the tutor, such as in the form of positive confirmation or praise.

- *Sensemaking*: A mode in which the tutor is helping the student gain conceptual understanding as opposed to procedural accuracy, e.g., explaining why a particular problem-solving step is appropriate in a particular circumstance.

- *Metacognitive Support*: a mode in which the tutor makes assertions to help the student think about the problem solving process at a “meta” level, e.g., the importance of perseverance, checking for careless errors, etc.

- *Rapport Building*: a mode in which the tutor or tutee or both intend primarily to build rapport, e.g., expressions of praise, apologies, affect queries (How are you this evening?).

- *Process Negotiation*: a mode in which the two conversational partners negotiate aspects of the tutorial process itself such as whether there is time to work on another problem, etc.

- *Session Summary*: A mode in which the tutor reviews the session, with an emphasis on what the student might have or ought to have learned.

- *WrapUp/Closing*: A mode in which the tutor moves to close out the session, often ending in an exchanges of farewells, and the tutor’s reminder to complete a satisfaction survey.

- *Off Topic*: A mode in which the students engages the tutor in an off-topic conversation.

It can be seen from Table 4 that the expert-defined *Modeling* mode is dominated by tutor assertions, student expressives, and tutor requests.

In order to better understand the relationship between the hidden states and the expert modes we have applied an automatic, optimal method to map each of the hidden states onto the 16-expert defined modes based on the dialogue act profiles, i.e. distribution of dialogue acts, of the hidden states and expert modes, as explained next.

An interesting outcome of this mapping step would be an understanding of what discovered hidden states map to which of the expert modes and also to identify which of the discovered state are not mapped and therefore may constitute novel modes, not identified previously in the literature.

Table 3. Interpretation of discovered modes based on the distribution of dialogue acts in the big dataset (S - Student, T - Tutor).

State	Interpretation	Dominant D-Acts
S1	Modeling/Explaining	T-Explanation(22.0822),T-Expressive(15.2577),T-Assertion(14.7110)
S2	Confidence-boosting Socratic fading	T-Expressive(48.1444),T-Question(35.4434), S-Expressive(10.3880)
S3	T feedback	T-Correction(27.6717),T-Assertion(26.0130),T-Request(17.7360)
S4	Affect-aware scaffolding or fading	T-Expressive(80.6667),T-Request(6.9660),T-Assertion(5.7497)
S5	Affective feedback	T-Expressive(62.3456),T-Confirmation(13.8997),T-Assertion(13.2127)
S6	S Request	S-Request(75.0016),S-Explanation(4.9769),S-Correction(4.5600)
S7	Almost total fading	S-Assertion(65.9099),S-Prompt(14.1390),S-Request(12.8968)
S8	T-asserting with affective S acknowledging	T-Assertion(47.6973),S-Expressive(28.2179),S-Assertion(18.4654)
S9	T-asserting and T-suggesting	T-Assertion(81.3236),T-Suggestion(7.1746),T-Expressive(6.1828)
S10	T answering questions	T-Expressive(56.5312),T-Answer(20.9574),T-Assertion(7.0612)
S11	Light-scaffolding	T-Prompt(78.8110),T-Expressive(6.2705),S-Assertion(3.3696)
S12	Self-Monitored Fading	S-Assertion(50.6311),T-Assertion(10.3381),T-Expressive(8.3451)
S13	S Eureka-gratitude	S-Expressive(76.6243),S-Assertion(6.2723),S-Request(5.3441)
S14	Opening	S-Question(94.6606),S-Expressive(2.5022),S-Request(1.3288)
S15	T-request	T-Request(88.2270),T-Prompt(8.6873),T-Assertion(1.8864)
S16	Heavy-scaffolding	T-Prompt(44.5824),T-Assertion(37.1729),S-Assertion(8.0842)
S17	S answering	S-Expressive(48.4172),S-Assertion(15.2621),S-Answer(10.9857)
S18	T clarifying and acknowledging	T-Assertion(34.6596),T-Request(24.0651),T-Confirmation(16.2986)
S19	S feedback	S-Expressive(36.1512),S-Confirmation(21.2304),S-Assertion(19.6898)

In order to map from hidden states to expert modes, we compute a hidden-state-to-expert-mode correspondence matrix based on the Information-Radius (IR; Niraula et al. 2013) distribution similarity measure. That is, we view each hidden state as a distribution over dialogue acts (i.e., the emission probabilities of the HMM) and also each expert-mode as a distribution over dialogue acts (we derived the distribution of dialogue acts from the human annotated data). We then compute the similarity between such distributions for all pairs of hidden states and expert modes which can be stored in a matrix. Once this state-to-mode similarity matrix is available, we apply the optimal matching algorithm (Kuhn, 1955; Munkres, 1957) to find the one-to-one mapping from hidden states to expert modes that maximizes the overall, global similarity between hidden states and expert modes such that one state is mapped to one and only one expert mode. Because we have more hidden states (19) then modes (16) we added 3 dummy expert modes which we assign to have a 0 similarity with any of the hidden states. The hidden states that are mapped to the dummy modes would be the states that indicate novel, previously unidentified modes. Table 5 shows the optimal mapping from the hidden states to the expert modes. It can be noted that states 5, 8, and 19 are not mapped to any of the expert modes, which we conclude as being novel dialogue modes.

Table 4. Dominant dialogue acts for modes in the human annotated dataset.

MODE	TOP 3 MOST DOMINANT ACTS IN THE MODE (relative frequency as %)
Fading	T-Expressive(25.0114) S-Assertion(17.4897) T-Confirmation(9.4818)
ProblemID	S-Assertion(27.7061) T-Request(15.4225) T-Expressive(10.7397)
Assessment	T-Request(32.8406) S-Assertion(17.1363) S-Confirmation(15.9815)
RapportBuilding	T-Expressive(34.5935) S-Expressive(21.7480) T-Question(14.6748)
Metacognition	T-Assertion(33.3333) S-Assertion(20.9564) S-Expressive(15.0492)
ProcessNegotiation	T-Request(23.8422) S-Expressive(12.6022) T-Expressive(12.2587)
Modeling	T-Assertion(54.2757) S-Expressive(11.5401) T-Request(5.9991)
MethodID	T-Assertion(25.8130) S-Assertion(15.8537) T-Request(8.4350)
Sensemaking	T-Assertion(18.0807) T-Request(14.4870) S-Assertion(10.7572)
WrapUp/Closing	T-Expressive(44.6147) S-Expressive(29.7122) T-Request(16.1792)
Opening	T-Expressive(61.0459) S-Expressive(30.3030) T-Suggestion(2.7859)
Scaffolding	S-Assertion(17.2543) T-Prompt(16.1523) T-Assertion(15.7473)
Telling	T-Assertion(64.4992) S-Expressive(10.5992) T-Explanation(5.3798)
OffTopic	S-Assertion(26.8817) S-Expressive(15.5914) T-Expressive(11.2903)
MethodRoadMap	T-Assertion(67.2026) S-Expressive(11.2540) T-Suggestion(6.7524)
SessionSummary	T-Assertion(50.0000) S-Expressive(11.5044) S-Assertion(9.2920)

Table 5. Optimal mapping from hidden states to expert modes using Information Radius and Optimal Matching algorithm.

State	Expert Mode
S1	SessionSummary
S2	MethodRoadMap
S3	SenseMaking
S4	Opening
S5	**NEW**
S6	MetaCognition
S7	OffTopic
S8	**NEW**
S9	Assessment
S10	MethodID
S11	Scaffolding
S12	ProcessNegotiation
S13	Modeling
S14	Telling
S15	RapportBuilding
S16	Fading
S17	WrapUp/Closing
S18	ProblemID
S19	**NEW**

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

We presented in this paper our work on automatically discovering dialogue modes from a large pool of human-to-human tutorial dialogues. The modes are mixtures of tutor and tutee actions in the form of hidden states discovered following the HMMs framework. Because these modes are characterized by distributions of tutor and tutee's actions derived from professional tutoring sessions, they can be used to dynamically check the quality of a tutoring session by dynamically building the profile of a tutoring session, e.g. managed by GIFT or another ITS, to indicate to what degree a particular tutoring session matches the profile of professional tutor's sessions in terms of dialogue modes.

ACKNOWLEDGMENTS

This research was partially sponsored by The University of Memphis, the Institute for Education Sciences (IES) under award R305A100875, and by a subcontract to The University of Memphis from Tutor.com under an award by Department of Defense.

REFERENCES

Austin, J.L. (1962). How to do Things with Words. Oxford University Press, 1962.

Boyer, K.E., Phillips, R., Ingram, A., Ha, E.Y., Wallis, M.D., Vouk, M.A., & Lester, J.C. (2011). The International Journal of Artificial Intelligence in Education (IJAIED), Vol. 21 No. 1, 2011, 65-81.

Cade, W., Copeland, J., Person, N., & D'Mello, S. (2008). Dialog modes in expert tutoring. Proceedings of the 9th International Conference on Intelligent Tutoring Systems, Montreal, Canada, 470-479.

Chi, M., VanLehn, K. & Litman, D. (2010). Do micro-level tutorial decisions matter: Applying reinforcement learning to induce pedagogical tutorial tactics. In V. Aleven, J. Kay & J. Mostow (Eds), Intelligent Tutoring Systems: 10th International Conference, ITS 2010 (pp. 184-193). Heidelberg, Germany: Springer.

Graesser, A., Person, N. and Magliano, J. (1995). Collaborative dialogue patterns in naturalistic one-to-one tutoring. Applied Cognitive Psychology 9, 495-522.

Heckerman, D. (1996). A tutorial on learning with bayesian networks. Technical report, Microsoft Research, 1996.

Jackson, G., Person, N. and Graesser, A. (2004). Adaptive tutorial dialogue in AutoTutor. Proc. Workshop on Dialog-based Intelligent Tutoring Systems at ITS’04.

Jeong, H., Gupta, A., Roscoe, R., Wagster, J., Biswas, G., & Schwartz, D. (2008). Using hidden markov models to characterize student behaviors in learning-by-teaching environments. In Intelligent Tutoring Systems: Vol. 5091. Lecture Notes in Computer Science (pp. 614-625). Montreal, Canada: Springer.

Kuhn, H.W. (1955). The Hungarian Method for the assignment problem. Naval Research Logistics Quarterly, volume 2:83–97.

Lafferty, J., Mccallum, A., & Pereira, F.C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of ICML, 282-289.

Li, C. and Biswas, G. (2002). A Bayesian Approach for Learning Hidden Markov Models from Data. Special issue on Markov Chain and Hidden Markov Models. Scientific Programming 10, 201–219 (2002).

Litman, D., & Forbes-Riley, K. (2006). Correlations between dialogue acts and learning in spoken tutoring dialogues. Natural Language Engineering, 12(2), 161-176.

Mitchell, T. (1997). Machine Learning, McGraw-Hill, 1997.Morrison, D. M., Nye, B. D., Samei, B., Datla, V. V., Kelly, C., and Rus, V. (2014). Building an Intelligent PAL from the Tutor. com Session Database-Phase 1: Data Mining (poster). In Proceedings of Educational Data Mining (EDM) 2014.

Morrison, D.C. & Rus, V. (2014). Is it a strategy or just a tactic?: A Martian perspective on the nature of human pedagogical dialogue. In Sottilare, R., Hu, X., Graesser, A. and Goldberg, B. (Eds.) Design Recommendations for Adaptive Intelligent Tutoring Systems: Adaptive Instructional Strategies, Volume II. Army Research Laboratory.

Munkres, J. (1957). Algorithms for the assignment and transportation problems. Journal of the Society for Industrial and Applied Mathematics, 5(1):32–38. Society for Industrial and Applied Mathematics.

Niraula, N., Banjade, R., Stefanescu, D., Rus, V. (2013). Experiments with Semantic Similarity Measures based on LDA and LSA. The 1st International Conference on Statistical Speech and Language Processing (SLSP 2013), July 29-31, Tarragona, Spain.

Ohlsson, S., DiEugenio, B., Chow, B., Fossati, D., Lu, X., and Kershaw, T.C (2007). Beyond the code-and-count analysis of tutoring dialogues. In AIED07, 13th International Conference on Artificial Intelligence in Education, Marina Del Rey, CA, July 2007.

Rabiner, L.R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proc. IEEE, 77 (1989).

Rus, V., Graesser, A., Moldovan, C., & Niraula, N. (2012). Automatic Discovery of Speech Act Categories in Educational Games, 5th International Conference on Educational Data Mining (EDM12), June 19-21, Chania, Greece.

Rus, V., Niraula, N., Maharjan, N., and Banjade, R. (2015). Automated Labelling of Dialogue Modes. Proceedings of the Florida Artificial Intelligence International Conference (FLAIRS-15).

Schwarz, G. (1978). Estimating the dimension of a model, Annuals of Statistics 6 (1978), 461–464.

Searle, J.R. (1969). *Speech acts: An essay in the philosophy of language* (Vol 626). Cambridge university press.

Sottilare, R., Brawner, K., Goldberg, B. & Holden, H. (2012). The Generalized Intelligent Framework for Tutoring (GIFT). US Army Research Laboratory.

ABOUT THE AUTHORS

Dr. Vasile Rus *is an Associate Professor of Computer Science with a joint appointment in the Institute for Intelligent Systems (ITS) whose areas of expertise are computational linguistics, artificial intelligence, software engineering, and computer science in general. His research areas of interest include question answering and asking, dialogue-based intelligent tutoring systems (ITSs), knowledge representation and reasoning, information retrieval, and machine learning. For the past 10 years, Dr. Rus has been heavily involved in various dialogue-based ITS projects including systems that tutor students on science topics (DeepTutor), reading strategies (iSTART), writing strategies (W-Pal), and metacognitive skills (MetaTutor). Currently, Dr. Rus leads the development of the first intelligent tutoring system based on learning progressions, DeepTutor (www.deeptutor.org). He has coedited two books, received several Best Paper Awards, and authored more than 100 publications in top, peer-reviewed international conferences and journals. He is currently Associate Editor of the International Journal on Artificial Intelligence Tools.*

Nabin Maharjan *is currently a PhD student in Computer Science at The University of Memphis working with Dr. Rus in the area of dialogue-based Intelligent Tutoring Systems and semantic processing. His research areas of interest include dialogue-based intelligent tutoring systems, computational semantics, information retrieval and machine learning.*

Rajendra Banjade *is currently a PhD student in Computer Science at The University of Memphis working in the area of dialogue-based Intelligent Tutoring Systems and semantic processing.*

Motivational Feedback Designs for Frustration in a Simulation-based Combat Medic Training Environment

Jeanine A. DeFalco[1], Ryan S. Baker[1], Luc Paquette[1], Vasiliki Georgoulas[1],
Jonathan Rowe[2], Bradford Mott[2,], James Lester[2]
[1]Teachers College, Columbia University
[2] North Carolina State University

INTRODUCTION

Tutoring systems researchers have recognized the need to identify and address affective states that lead to disengagement in learning activities (Baker, D'Mello, Rodrigo, & Graesser, 2010; D'Mello, Lehman, & Graesser, 2011; D'Mello Strain, Olney, & Graesser, 2013; Forbes-Riley, Litman, Friedberg, 2011; Gee, 2004, 2007; Picard et al., 2004). Some affective states have relatively uncomplicated relationships with student learning outcomes – engaged concentration appears to be positively associated while boredom is negatively associated (Craig et al., 2004; Pardos et al., 2014). The affective state of frustration is more complex. Liu and colleagues (2013) have found that brief frustration is not problematic, but that extended frustration is associated with worse learning outcomes. Understanding how intelligent tutoring system can respond to frustration is likely to be an important aspect of future affect-sensitive learning environments (Picard et al., 2004).

In this paper, we discuss potential intervention designs that will be used in upcoming research on how intelligent learning environments can respond to student frustration. These designs will be embedded in the context of the GIFT architecture (Sottilare, Goldberg, Brawner, & Holden, 2012), designed for the context of using the TC3 courseware and vMedic serious game (Sotomayor, 2010) focusing specifically on game-based training materials for hemorrhage control. Three designs will be presented in this paper; the designs are informed by three theories on learner motivation, specifically control-value theory (Pekrun, 2006), social identity theory (Tajfel & Turner, 1979), and theory of self-efficacy (Bandura, 1977). By studying designs connected to three distinct theoretical paradigms, we can investigate which theoretical paradigm is most useful for driving the design of adaptations to frustration. Adapting to learner frustration depends on knowing which students are frustrated. To this end, this project builds on prior work on the detection of affect in vMedic. In this work, baseline data was collected in September 2013 on learner engagement and affect while trainees were learning about hemorrhage control through vMedic. This baseline data was used to develop affect detectors for frustration using both interaction-based and posture-based approaches (Paquette et al., accepted).

The next task in these efforts is to design intervention messages aimed at improving trainee engagement and learning outcomes, leveraging the information provided by automated detection of affect. As such, we will examine the impact of motivational feedback messages, incorporated into vMedic.

Developing frustration feedback interventions, and studying their impact on learning, will contribute to a greater understanding of the relationship between affect, engagement, and learning outcomes, and how negative affect can be addressed by automated systems. In the long term, this effort will help us understand how to design affect-sensitive tutoring systems, realized within the GIFT architecture.

THEORY AND PREVIOUS RESEARCH

Frustration and learning

As discussed above, the relationship between frustration and engagement is complex. While negative relationships between frustration and learning are not always seen (e.g. Craig et al., 2004; Pardos et al., 2014), some studies have suggested that this is because the duration of frustration matters more than its absolute incidence (e.g. Liu et al., 2013). Beyond just learning, frustration has been found to divert student attention from learning tasks and lead the learner to worry excessively about failure (Kapoor, Burleson, & Picard, 2007; McQuiggan, Lee, & Lester, 2007). In the specific context of vMedic, unpublished early research suggests that frustration is negatively correlated with learning outcomes, making it important to study in this context.

The dynamic nature of frustration – where brief frustration can yield positive learning gains but sustained frustration yields negative outcomes (Liu et al., 2013), a pattern also seen for the related affective state of confusion (Lee et al., 2011; Liu et al., 2013; D'Mello, Lehman, Pekrun, & Graesser, 2014) – requires interventions that are timely.

Motivating the learner to persist through frustration

Appropriately responding to student frustration depends on understanding the nature of the type of intervention that will be utilized. In this paper, we focus on feedback messages, which are easy to implement in the GIFT architecture, and which can be generalized with relative ease within that architecture. Narciss (2008) notes that motivational feedback can be conducted with many purposes, one of which is to sustain effort and persistence in the learning task. This feedback model contextualizes feedback within the theories of self-regulated learning where the primary function of feedback is guiding the learner to successfully regulate their learning process (Butler & Winne, 1995; Narciss, 2008). Narciss (2008) maintains that feedback that guides learners to successful task completion through motivating them rather than immediately providing answers of correct responses can provide a learner with a mastery experience, leading to the development of positive self-efficacy (Bandura, 1997; Usher & Pajares, 2006).

Designing computational systems that can both recognize when a learner is frustrated provide an effective intervention, which positively impacts the learner’s future actions and their learning outcomes, is a complicated process. Thus far, research in alleviating frustration through feedback messages has achieved mixed results (Klein, Moon, & Picard, 2002; Robison, McQuiggan, & Lester, 2009), emphasizing the importance of selecting interventions that have low probability of negative consequences to learners (Robison et al., 2009). As such, it is important to attempt to deliver motivational feedback for frustration at the right time, and to select interventions that have limited cost if incorrectly applied.

PROJECT DESIGN: FRUSTRATION FEEDBACK INTERVENTIONS

Within GIFT, motivational feedback messages will be authored as both a text and an audio message to be delivered by an embodied pedagogical agent once the frustration detectors, built into GIFT, detect frustration of the trainee. We will compare three types of motivational feedback, plus a control condition, making a total of four conditions: one condition providing messages designed according to control-value theory, one condition providing messages designed according to social-identity theory, one condition providing messages designed according to self-efficacy theory, and a control condition with no feedback messages. For each of the feedback conditions, a separate message will be authored for each of the four scenarios the trainees will complete while engaged with vMedic. These messages will be delivered via the GIFT architecture, appearing to be given by a pedagogical agent (cf. Klein et al., 2002).

Control-value Theory

One path to intervening on frustration involves framing feedback messages within the context of control-value theory (Pekrun, 2006). The objectives of this feedback are to 1) seek to motivate learners to persist in the learning activity based on an implicit appeal to the learner's perceived controllability of achievement activities and their outcomes, as well as 2) highlight the value and importance of the learning activities and outcomes (Artino, Holmboe, & Durning, 2012). Control-value theory was developed by Pekrun (, 2006) as a comprehensive, integrative approach to understanding emotions in education. When individuals feel in or out of control of achievement activities and outcomes that are subjectively important to them, they experience specific achievement emotions (Pekrun, Frenzel, Goetz, & Perry, 2007). Achievement activities are mediated by emotions that influence cognitive resources, motivation, strategy choices, and intrinsic and extrinsic regulation of learning. The outcome of these achievement activities in turn influences students' emotions (Pekrun, Frenzel, Goetz, & Perry, 2007).

Control-value theory, then, implies that student achievement emotions such as frustration can be influenced by changing the student's subjective perception of control and value through a shaping of the learning environment (Pekrun, Frenzel, Goetz, & Perry, 2007; Kim, 2010). In the specific case of this study, we will influence trainees' perception of control and value using feedback messages that include facts pulled from journal papers on the effectiveness of using relevant medical procedures that can be applied in the field such as tourniquets for hemorrhage control, suggesting that participants can control casualty outcomes through their actions, creating positive outcomes that they value (survival of a fellow soldier). An example of a feedback message in this condition includes the following: "Studies have shown that between 17% - 19% of deaths in Vietnam could have been prevented if tourniquets had been used," (DePillis, 2013).

Social Identity Theory

A second path to intervening on frustration involves framing feedback messages to highlight the trainee's role as a member of a group, in this case, as a member of the United States Army. This design capitalizes on social identity theory, which states that our identities are formed in large part through the groups to which we belong, creating some degree of uniformity of perception and action among group members (Stets & Burke, 2000). While the authors have not identified literature on using social identity feedback messages to address frustration in tutoring systems, social identity theory has been used to motivate human-human training to shape behavior and decision-making, including attitudes and value-orientations – particularly in the education and training of military cadets at West Point (Franke, 1997; Franke, 2000). The existing use of this approach in military training highlights its potential value for automated adaptation designed for this population.

Shamir, House, and Arthur (1993) have argued that leaders strengthen social identification through the use of cultural symbols such as slogans, symbols, rituals, and ceremonies that highlight collective identity, superiority, and uniqueness. Taking into consideration, then, the relationship between cultural slogans and a soldier's social identity, the second feedback condition was chosen in the form of quotes by former Generals, identifying the learner as a "soldier" and calling on the learner's identification as a member of the US Army. These identity-based motivational feedback messages highlight how to handle frustration, and the importance of persistence in the face of frustration. The messages are connected to military leaders in order to capitalize on the notion that people prefer actions that are identity-congruent (Oyserman & Destin, 2010). An example of a feedback message in this condition includes the following: " 'Duty, Honor, Country' — those three hallowed words reverently dictate what you ought to be, what you can be, what you will be.' - General Douglas MacArthur, Jr." (MacArthur & Westmoreland, 1964).

Self-Efficacy Theory

A third path to intervening on frustration involves framing feedback messages based on the theory of self-efficacy, directed at the learner as an individual, and their ability to succeed in the task if they persist. Self-efficacy is known to correlate positively to academic performance and persistence rates (Bong, 2001; Kaun & Nauta, 2001; Multon, Brown, & Lent, 1991; Wood & Locke, 1987) – this intervention will test if it can address frustration as well.

Bandura's (1986) socio-cognitive perspective on the role of self-efficacy theorizes that individuals are proactive and self-regulating rather than merely reactive and controlled by biological or environmental forces. Bandura's social cognitive theory (1997, 2002) argues that perceived self-efficacy influences a person's motivation for tasks, actions towards goal achievement, perseverance on tasks, and responses to failures.

For the purposes of using the self-efficacy construct to inform the design of feedback messages, the goal is to design feedback that persuades the learner they have the necessary skills to succeed. As such, the feedback messages informed by the theory of self-efficacy will be designed to support the trainee's belief that they can succeed in the system and attain their learning goals while engaged in vMedic. An example of a feedback message in this condition includes the following: "Your best outcomes will be achieved if you persist."

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

In this article, we discuss our efforts to design feedback that can address trainee frustration, within the context of vMedic. We articulate three potential designs for feedback, based on three relevant theories:

control-value theory, social-cognitive theory, and self-efficacy theory. We will investigate the impact of these interventions through a study where these feedback messages are delivered by a pedagogical agent embedded in the GIFT architecture, executed within the vMedic training system. The findings of this study, it is hoped, will shed light on how to develop affect-responsive tutoring systems for U.S. Army personnel. By creating online training that is sensitive to trainee affect, and helping trainees learn to regulate their behavior better in frustrating situations, we can take a step towards online training that better prepares U.S. Army soldiers for the many challenges they will face.

REFERENCES

Artino, A., Holmboe, E., & Durning, S. (2012) Control-value theory: Using achievement emotions to improve understanding of motivation, learning, and performance in medical education: AMEE Guide No. 64. *Medical Teacher*. 34: 148-160.

Baker, R.S.J.d., D'Mello, S.K., Rodrigo, M.M.T., Graesser, A.C. (2010). Better to Be Frustrated than Bored: The Incidence, Persistence, and Impact of Learners' Cognitive Affective States during Interactions with Three Different Computer-Based Learning Environments. *International Journal of Human-Computer Studies*, 68 (4), 223-241.

Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavior change. *Psychological Review*, 84, 191–215.

Bandura, A. (1986). *Social foundations of thought and action: A social cognitive theory*. Englewood Cliffs, NJ: Prentice Hall.

Bandura A. (1997). *Self-efficacy: The exercise of control*. New York, NY: W.H. Freeman and Company.

Bandura, A. (2002). Social cognitive theory in cultural context. *Applied Psychology*, *51*,(2), 269-290

Bong, M. (2001). Role of self-efficacy and task-value in predicting college students' course performance and future enrollment intentions. *Contemporary Educational Psychology*,*26*(4), 553-570

Butler, D. L., & Winne, P. H. (1995). Feedback and self-regulated learning: A theoretical synthesis. *Review of educational research*, *65*(3), 245-281

Craig, S. D., Graesser, A. C., Sullins, J., & Gholson, B. (2004). Affect and learning: an exploratory look into the role of affect in learning. *Journal of Educational Media*, 29, 241-250.

D'Mello, S. K., Lehman, B., & Graesser, A. (2011). A motivationally supportive affect-sensitive AutoTutor. In *New perspectives on affect and learning technologies* (pp. 113-126). Springer New York.

D'Mello, S., Lehman, B., Pekrun, R., & Graesser, A. (2014). Confusion can be beneficial for learning. *Learning and Instruction*, *29*, 153-170.

D'Mello, S. K., Strain, A. C., Olney, A., & Graesser, A. (2013). Affect, meta-affect, and affect regulation during complex learning. In *International handbook of metacognition and learning technologies* (pp. 669-681). Springer New York.

DePillis, L. (2013). The return of the tourniquet: What we learned from war led to lives saved in Boston. *New Republic*, April 17, 2013. Retrieved on February 8, 2015 at http://www.newrepublic.com/article/112939/boston-marathon-bombing-return- tourniquetForbes-Riley,

Forbes-Riley, K., Litman, D., & Friedberg, H., (2011). Eds. Rafael Calvo and Sidney D'Mello In New Perspectives on Affect and Learning Technology, 3: 169-181.

Franke, V. (1997). Warriors for peace: The next generation of military leaders. *Armed Forces & Society*, 24: 33-59.

Franke, V. (2000). Duty, honor, country: The social identity of West Point cadets. *Armed Forces & Society*, *26*(2): 175-202.

Gee, J. P. (2004). Learning by design: Games as learning machines. *Interactive Educational Multimedia*, (8), 15-23.

Gee, J. P. (2007). *Good video games+ good learning: Collected essays on video games, learning, and literacy*. New York: P. Lang.

Kapoor, A., Burleson, & W., Picard, R. (2007). Automatic Prediction of Frustration. *International Journal of Human Computer Studies*, 65(8), 724-736.

Kaun, J., & Nauta, M. (2001). Social-cognitive predictors of first-year college persistence: The importance of proximal assessment. *Research in Higher Education, 42*, 633-652

Kim, C. (2010). The role of affective and motivational factors in designing personalized learning environments. *Education Tech Research Dev.,* 60:563-584. DOI 10.1007/s11423-012-9253-6

Klein, J., Moon, Y., & Picard, R. W. (2002). This computer responds to user frustration: Theory, design, and results. *Interacting with computers*, *14*(2), 119-140.

Lee, D. M., Rodrigo, M. M., Baker, R. S. J. d., Sugay, J., & Coronel, A. (2011). Exploring the Relationship Between Novice Programmer Confusion and Achievement. In *Proceedings of the 4th bi-annual Inte*

Liu, Z., Pataranutaporn, V., Ocumpaugh, J., Baker, R.S.J.d. (2013) Sequences of Frustration and Confusion, and Learning. *Proceedings of the 6th International Conference on Educational Data Mining*, 114-120.

MacArthur, D., & Westmoreland, W. C. (1964). *Duty, honor, country*. Armed Forces Information and Education, Department of Defense.

McQuiggan, S. W., & Lester, J. C. (2006). Diagnosing self-efficacy in intelligent tutoring systems: an empirical study. In *Intelligent Tutoring Systems* (pp. 565-574). Springer Berlin Heidelberg.

Multon, K. D., Brown, S. D., & Lent, R. W. (1991). Relation of self-efficacy beliefs to academic outcomes: A meta-analytical investigation. *Journal of Counseling Psychology*, **38**, 30-38.

Narciss, S. (2008). Feedback strategies for interactive learning tasks. *Handbook of research on educational communications and technology*, 3, 125-144.

Oyserman, D. & Destin, M. (2010). Identity-based motivations: Implications for intervention. *The Counseling Psychologist*, 38(7): 1001-1043.

Paquette et al. (accepted). Sensor-Free or Sensor-Full: A Comparison of Data Modalities in Multi-Channel Affect Detection. *Proceedings of the 8th International Conference on Educational Data Mining*.

Pardos, Z.A., Baker, R.S., San Pedro, M.O.C.Z., Gowda, S.M., Gowda, S.M. (2014) Affective states and state tests: Investigating how affect and engagement during the school year predict end of year learning outcomes. *Journal of Learning Analytics*, 1 (1), 107-128.

Pekrun R. (2006). The control-value theory of achievement emotions: Assumptions, corollaries, and implications for educational research and practice. *Education Psychology Rev*, 18, 315–341.

Pekrun, R., Frenzel, A., Goetz, T., & Perry, R. (2007). The control-value theory of achievement emotions: An integrative approach to emotions in education. *Emotion in education*, (Ed) Paul A. Schutz & Reinhard Pekrun. Amsterdam: Academic Press.

Picard, R., Papert, S., Bender, W., Blumberg, B., Breazeal, C., Cavallo, D., Machover, T., Resnick, M., Roy, D., & Strohecker, C. (2004). Affective learning – a manifesto. *BT Technology Journal*, 22(4): 253-269.

Robison, J., McQuiggan, S., & Lester, J. (2009). Evaluating the consequences of affective feedback in intelligent tutoring systems. *Proceedings of the Affective computing and intelligent interaction workshops ACII* 2009.

Shamir, B., House, R.J., & Arthur, M.B. (1993). The motivational effects of charismatic leadership: A self-concept based theory. *Organization Science*, *4*(4), 577-594.

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

Stets, J., & Burke, P. (2000). Identity theory and social identity theory. *Social Psychology Quarterly* 63 (3): 224-237

Tajfel, H. and J.C. Turner. 1979. An integrative theory of intergroup conflict. In *The social psychology of intergroup relations,* ed. S. Worchel and W.G. Austin, 33–47. Chicago: Nelson-Hall.

Sotomayor, T. M. (2010). Teaching tactical combat casualty care using the TC3Sim game-based simulation: a study to measure training effectiveness. *Studies in health technology and informatics*, *154*, 176-179.

Usher, E. L., & Pajares, F. (2008). Sources of self-efficacy in school: Critical review of the literature and future directions. *Review of Educational Research, 78*(4), 751-796.

Wood, R. & Locke, E. (1987). The relation of self-efficacy and grade goals to academic performance. *Educational and Psychological Measurement, 17,* 1013-1024.

ABOUT THE AUTHORS

Jeanine A. DeFalco *is a member of Dr. Baker's Lab and a Doctoral Research Fellow in the Department of Human Development, Cognitive Studies at Teachers College, Columbia University. Jeanine's research focuses on motivation, engagement, and instructional design in face-to-face and tech-based learning platforms. Jeanine holds an MA in Education from New York University, and a Masters in Drama Studies from The Johns Hopkins University.*

Dr. Ryan Baker *is Associate Professor of Cognitive Studies at Teachers College, Columbia University, and Program Coordinator of TC's Masters of Learning Analytics. He earned his Ph.D. in Human-Computer Interaction from Carnegie Mellon University. Dr. Baker was previously Assistant Professor of Psychology and the Learning Sciences at Worcester Polytechnic Institute, and served as the first technical director of the Pittsburgh Science of Learning Center DataShop. He is currently serving as the founding president of the International Educational Data Mining Society, and as associate editor of the Journal of Educational Data Mining. His research combines educational data mining and quantitative field observation methods to better understand how students respond to educational software, and how these responses impact their learning.*

Dr. Luc Paquette *is a post-doctoral research associate working with Dr. Ryan Baker in the department of Human Development at Teachers College, Columbia University. His current research focuses on integrating cognitive modeling and educational data mining approaches for the modeling of the disengaged behaviour of students who "Game the System". He is also involved in the development of interaction-based detectors of pedagogically relevant affective states. Luc Paquette holds a PhD in Computer Science from the University of Sherbrook.*

Vasiliki Georgoulas*, is a Research Psychologist for the United States Military Academy and a Doctoral Research Fellow in the Department of Human Development at Teachers College, Columbia University. Vasiliki's research focuses on cognitive processing and psychological resilience, soldier performance in combat, and other dangerous contexts in the Army.*

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. Bradford Mott *is a Senior Research Scientist in the Center for Educational Informatics at North Carolina State University. His research interests include artificial intelligence and human-computer interaction, with applications in educational technology. In particular, his research focuses on game-based learning environments, intelligent tutoring systems, and computational models of interactive narrative. He has many years of software development experience from industry, including extensive experience in the video game industry, having served as Technical Director at Emergent Game Technologies where he created cross-platform middleware solutions for Microsoft's Xbox and Sony's PlayStation video game consoles.*

Dr. James Lester *is a 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. The recipient of a National Science Foundation CAREER Award, he has been named a AAAI Fellow by the Association for the Advancement of Artificial Intelligence.*

Toward a Modular Reinforcement Learning Framework for Tutorial Planning in GIFT

Jonathan Rowe[1], Bradford Mott[1], James Lester[1], Bob Pokorny[2], Wilbur Peng[2], and Benjamin Goldberg[3]
North Carolina State University[1], Intelligent Automation, Inc.[2], U.S. Army Research Laboratory[3]

INTRODUCTION

Intelligent tutoring systems (ITSs) are highly effective at fostering learning gains across a broad range of educational domains (VanLehn, 2011; Woolf, 2008). Tutorial planning is a critical component of ITSs, controlling how scaffolding is structured and delivered to learners. Tutorial planners operate at multiple levels, including the macro-level (e.g., selecting problems for learners to solve) and micro-level (e.g., delivering tailored hints about specific problems). Devising computational models that scaffold effectively—determining when to scaffold, what type of scaffolding to deliver, and how scaffolding should be realized—is a critical challenge for the field.

Simulation-based training is an especially challenging (yet promising) environment for tutorial planning (McAlinden, Gordon, Lane & Pynadath, 2009). Many simulation-based virtual training environments feature open-ended scenarios with multiple solutions, numerous problem-solving paths, and complex dynamics. Modeling learning in complex virtual training environments is also marked by inherent uncertainty, demanding the use of probabilistic representations that account for the likelihood of alternate learner behaviors. Although these challenges are significant, designing effective tutorial planners for complex virtual training environments holds great potential because of the significant potential to create highly effective learning environments that simultaneously model complex real-world problem scenarios and personalize guidance to individual learners.

Tutorial planners often suffer from several significant limitations. First, creating tutorial planners is expensive, requiring labor-intensive knowledge engineering processes that involve close collaboration between subject matter experts, education experts, and software developers (Murray, 2003). Second, once a tutorial planner has been created, it typically remains fixed; it does not improve or change over time unless manually updated by an expert. Third, tutorial planners often model rules for scaffolding by using symbolic representational techniques, which are poorly suited for reasoning under uncertainty. In recent years several ITS research labs have begun to investigate methods for devising data-driven tutorial planners that automatically induce scaffolding models from corpora of learner data (Chi, VanLehn & Litman, 2010; Rowe & Lester, in press). Leveraging decision-theoretic frameworks, such as Markov decision processes (MDPs), these models explicitly account for the inherent uncertainty in how learners respond to different types of tutorial strategies and tactics, and automatically induce tutorial planning policies in order to optimize measures of learning outcomes. Yet, there are important open questions to be addressed regarding the generalizability and scalability of these approaches across different domains, populations, and educational settings.

To begin to address these questions, we are embarking on a research collaboration involving three complementary teams from North Carolina State University (NCSU), Intelligent Automation, Inc. (IAI), and the U.S. Army Research Laboratory (ARL) to investigate how to devise data-driven tutorial planning models that automatically improve their instructional techniques, strategies and tactics as learners interact with a virtual training environment. Our team will utilize, and substantially extend, the Generalized Intelligent Framework for Tutoring (GIFT) to incorporate support for modeling tutorial planners as MDPs. This will enable the creation of tools for automatically inducing scaffolding policies from learner data using *modular reinforcement learning*. Modular reinforcement learning is a multi-goal extension of classical single-agent reinforcement learning. It involves decomposing a planning task into multiple concurrent sub-problems, encoded as MDPs. Each MDP is solved separately, and then re-combined to control the intelligent system's overall behavior, with arbitration amongst solution policies as they come into conflict. The project will build upon NCSU's work on data-driven tutorial planning in game-based learning environments, as well as IAI's work on assessment and scaffolding for complex simulation-based training environments. In the remainder of this paper, we describe our modular reinforcement learning framework for tutorial planning, plans to devise tutorial planning models for a counterinsurgency and stability operations training environment, and recommendations for enhancements to GIFT that will facilitate data-driven tutorial planning.

DATA-DRIVEN TUTORIAL PLANNING

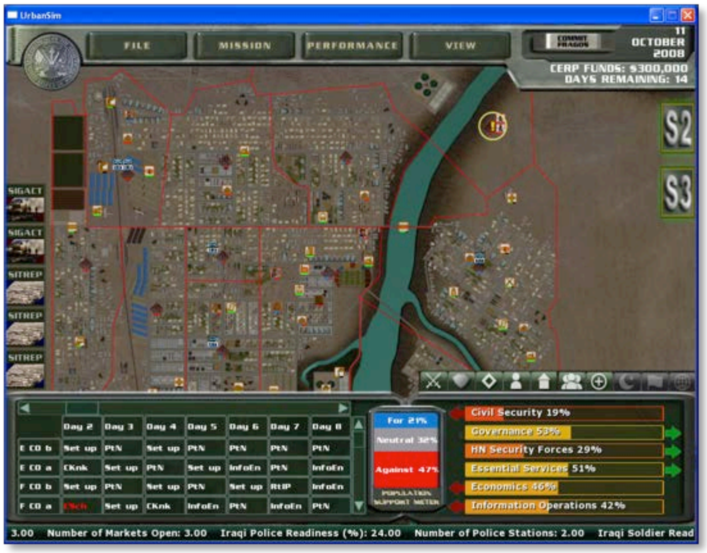
Data-driven methods hold considerable promise for addressing the challenges of tutorial planning. We propose to model the task of automatically inducing and refining a data-driven tutorial planner as a modular reinforcement learning problem. Reinforcement learning refers to a class of machine learning techniques that involve acting 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. In contrast to supervised machine learning, the agent is not provided with external instruction about which actions to take. Instead, the environment produces rewards that provide positive or negative feedback about the agent's actions. 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 techniques have been the subject of growing interest in the intelligent tutoring systems community (Barnes & Stamper, 2008; Beck, Woolf, & Beal, 2000; Chi, VanLehn, & Litman, 2010). This work has emphasized probabilistic models of behavior, as opposed to explicit models of cognitive states, in order to analyze student learning. For example, Chi, VanLehn and Litman (2010) used MDPs to model tutorial dialogues, devising pedagogical tactics directly from student data in the Cordillera physics tutor. Barnes and Stamper (2008) modeled students' logic proof sequences as MDPs in order to automatically generate context-appropriate hints. Complementary work investigating partially observable Markov decision processes (POMDPs) to model tutorial planning has been explored, yielding novel approaches for compactly representing MDP state representations (Brunskill & Russell, 2011; Folsom-Kovarik, Sukthankar & Schatz, 2013). In our work, we focus on inducing tutorial planning models directly from learner data, similar to the approach taken by Chi, VanLehn and Litman (2010). We aim to devise generalized tutorial planning models that control a broad range of scaffolding decisions in a complex simulation-based training environment, UrbanSim.

UrbanSim Simulation-Based Training Environment

UrbanSim (See Figure 1) is an open-ended simulation-based virtual training environment for counterinsurgency and stability operations. In UrbanSim, learners act as a battalion commander whose mission is to maximize civilian support for the host nation government (McAlinden et al., 2009). 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. During each turn, UrbanSim presents (1) situation reports, such as "the Mayor is pleased with the increased electrical power available to citizens," (2) significant events, such as "an IED exploded at the Gas Station on Hwy2," and (3) civilian support for the host nation government, visually rendered from an overhead view using game engine technologies.

To inform the design of tutorial planners for complex simulation-based training environments such as UrbanSim, it is necessary to identify concepts and performance characteristics that are the targets of scaffolding and instructional remediation. In prior work, IAI applied a cognitive task analysis method to identify the performance patterns of trainees that should become targets for performance improvement. The task analysis took performance data of real learners, presented it in a format that was readily understandable to humans, and had experts (a) assign scores reflecting overall proficiency and (b) critique learners' actions. Their critiques ranged broadly across different topics, from what structures were repaired, to whom U.S. forces held meetings with, to security actions taken against specific threats. Analysts, with the assistance of subject matter experts, then characterized these comments into a relatively limited number of scoring rules. Learners' actions either followed good practice or violated good practice. When learner performance complied with the good practice, points were assigned to the learner. When learners violated good practice, points were deducted. Pokorny, Haynes, and Gott (2010) reported that this task analysis method yielded scores with excellent psychometric properties. The scores from experts and from automated scoring systems were valid, as they correlated with time of service in the job. This task analysis method was also reliable, as experts' scores



	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8
E CO b	Set up	PtN	Set up	PtN	PtN	PtN	PtN
E CO a	CKnk	Set up	PtN	Set up	InfoEn	PtN	InfoEn
F CO b	Set up	PtN	Set up	PtN	Set up	RtIP	InfoEn
F CO a	[illegible]	Set up	CKnk	InfoEn	PtN	InfoEn	PtN

Figure 1. UrbanSim simulation-based training environment

correlated well with other experts' scores, despite high apparent task complexity. Further, scores developed from scoring rules correlated well with experts' scores. Scores of experts across a variety of scenarios were all significant, with variations across different scenarios. For UrbanSim, violations of good practice were categorized into six categories: 1) security, 2) meetings, 3) support of the government, 4) information operations, 5) infrastructure selection, and 6) consistency over turns.

After devising the assessment rules noted above, which identified key targets of learner performance, instructional interventions were designed. To illustrate, consider the example of a rule stating that the learner should arrest a sniper. Learners gain points when they arrest the sniper, and lose points for the continued activity of the sniper. Instruction regarding the eradication of known insurgent groups would provide scaffolds for understanding the value in arresting the sniper. For example, a first hint regarding the sniper would ask the learner if there are threats to soldiers that the commander can easily target. If the learner arrests the sniper on the next action, the learner is commended for that action. If the learner does not arrest the sniper, a more directive hint is given to the learner: the sniper presents a threat to soldiers and government forces. The sniper should be removed from the environment according to the rules of law. If the learner allows the sniper to persist, an instructional message indicates that the sniper presents a danger to troops, and intelligence reports that the sniper's whereabouts is in the Musalla Quarter. The sniper can be found and arrested there. If the student does not arrest the sniper, but instead assassinates him, another instruction tells the learner that killing even a sniper encourages lawlessness in a society that needs examples of following due process.

The structure utilized to provide instruction that adapted to students' weaknesses was the Process Observer described by Lesgold and Nahemow (2001). In our version of the Process Observer, a left-most table column identifies violations of good practice. These might occur anytime during a scenario. Table columns to the right of the violation contain instructional messages presented to the student. The instruction we developed for UrbanSim contained three kinds of messages. One type of message was delivered if the student took an action in the game that was so egregious, that it indicated the player was not taking the game seriously. For example, if the player assassinated the Mayor, the player would receive an immediate recommendation to play in accordance with civil guidelines. A second type of message presented a post-problem reflection, also known as an After Action Review (AAR). The AAR directed the student to attend to all dimensions of performance. A third type of instructional message identified the student's performance that most violated good practices, and presented an instructional intervention tied to identified performance weakness. We used the scoring rules to identify the aspect of performance that most severely violated good performance. Instructional remediations were targeted at those aspects of performance that led to experts' most significant point violation. On specific turns of the game, we presented instructional interventions that targeted the student's current worst performance. If the instructional intervention on the target worked, then that intervention would have had a higher positive effect on scores of overall student quality than other interactions. In our instruction, we initiated these instructional interventions on Turns 3, 7 and 12.

We plan to build upon this foundation in the proposed project, focusing on the "security" and "meetings" dimensions of learner performance. Specifically, we plan to design, develop, evaluate, and iteratively refine a data-driven tutorial planner for UrbanSim that will scaffold trainee's learning processes within the security and meetings performance categories. Decisions about what types of scaffolding to deliver, and when to deliver them, will be induced and refined using modular reinforcement learning, rather than solely defined through manual authorship.

Modular Reinforcement Learning Framework for Tutorial Planning

We formalize tutorial planning as a modular reinforcement learning problem. Modular reinforcement learning is a multi-goal extension of classical single-agent reinforcement learning (Bhat, Isbell, & Mateas, 2006; Karlsson, 1997). In reinforcement learning, an agent learns a policy for selecting actions in an uncertain environment, guided by delayed rewards, in order to accomplish a goal (Sutton & Barto, 1998). The agent utilizes an environment-based reward signal in order to learn a policy, denoted π, which maps observed states to actions and maximizes total accumulated reward. Agents in reinforcement learning problems are typically modeled with Markov decision processes (MDPs).

Modular reinforcement learning tasks are formally defined in terms of *N* concurrent MDPs, $M = \{\mathbf{M_i}\}_{\mathbf{1}}^{\mathbf{N}}$, where each $M_i = (\mathbf{S_i}, \mathbf{A_i}, \mathbf{P_i}, \mathbf{R_i})$, 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, $\boldsymbol{\pi}^* = \{\boldsymbol{\pi}_\mathbf{i}^*\}_{\mathbf{1}}^{\mathbf{N}}$, where $\boldsymbol{\pi}_\mathbf{i}$ is the optimal policy for the constituent MDP M_i. Whenever two policies $\boldsymbol{\pi}_\mathbf{i}$ and $\boldsymbol{\pi}_\mathbf{j}$ with $i \neq j$ recommend different actions in the same state, an arbitration procedure must be applied.

Tutorial planning in simulation-based virtual training environments is naturally represented as a modular reinforcement learning problem: state consists of the learner's state and history as well as the learning environment's; actions represent the pedagogical decisions the planner can perform; a probabilistic state transition model encodes how learners, and the learning environment, respond to the planner's tutorial decisions; and a reward model encapsulates measures of trainees' learning outcomes, which the tutorial planner seeks to optimize. The solution to a modular reinforcement-learning problem is a set of policies, or mappings between states and tutorial actions, that govern how the tutorial planner scaffolds trainees' learning. If two policies conflict, externally defined arbitration procedures specify which policy prevails.

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. To perform this decomposition, we employ the concept of an adaptable event sequence (AES), an abstraction for a series of one or more instructionally related events that, once triggered, can unfold in several different ways within the learning environment (Rowe, 2013). To illustrate the concept of an AES, consider the earlier example of a sniper in UrbanSim. As suggested previously, the tutorial planner can intervene in one of several ways: 1) providing a high-level hint to the learner about potential threats in the area; 2) providing a mid-level hint about the nearby sniper, who should be removed; 3) providing a ground hint about the sniper's location, and a recommendation to arrest him; or 4) not intervening at all. Each of these four responses is an alternate pedagogical action related to the sniper. Each provides a distinct level of problem-solving support, which could be varyingly deployed based on the learner's performance and the state of the training environment. Moreover, decisions about what type of hint to deliver might occur just once during a training interaction, or multiple times over the course of several turns. 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).

AESs are not restricted to decisions about hints. In fact, they can encode a broad range of scaffolding types at both the macro- and micro-adaptive levels. A *Prompt-Explanation* AES may involve selecting whether to prompt a student to self-explain his problem-solving strategy and actions. An *Embedded-Assessment* AES may involve selecting whether to deliver a brief quiz during training to obtain a formative assessment of learners' knowledge. A *Select-Problem* AES may involve choosing whether the next problem scenario should emphasize a new set of knowledge and skills, or provide remedial practice on the current set of skills. In our framework, each AES is modeled separately as a Markov decision process (MDP), and tutorial decisions about what types of strategies and tactics to deploy are determined through reinforcement learning. Further, multiple AESs can be interleaved. A pedagogical decision about the sniper hint might be followed by a decision about presenting an embedded assessment, which could be followed by a successive decision about the sniper. AESs encode distinct threads of tutorial events, each potentially involving multiple decision points spanning an entire learning interaction. For this reason, AESs are sequential and operate concurrently. Each AES is modeled separately as a MDP, and tutorial decisions about scaffolding are determined through modular reinforcement learning.

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. Modular reinforcement learning is applied as follows. 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 tailored to the AES using manual or automatic feature selection techniques. Rewards, R_i, can be calculated from formative or summative assessments of student learning, such as a post-test. A state transition model P_i encodes the probability of transitioning between two specific states during successive decision points for the AES. To estimate the values of these parameters, we can collect training data from learners by deploying a tutorial planner that selects actions randomly, in effect sampling the space of tutorial policies and rewards (Chi, VanLehn, & Litman, 2010; Rowe & Lester, in press). Leveraging this 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. Specifically, we utilize dynamic programming methods (e.g., value iteration) to compute solution policies for each MDP using estimates of the state transition model and reward model inferred from the training corpus (Chi, VanLehn, & Litman, 2010; Sutton & Barto, 1998). In cases where two policies conflict, we utilize greatest mass arbitration, a domain-independent arbitration procedure that selects the action with the largest Q-value calculated during policy induction (Bhat, Isbell, & Mateas, 2006; Karlsson, 1997). In combination, this formulation provides a method for conceptualizing tutorial planning as an instance of modular reinforcement learning. Prior work with a narrative-centered learning environment for middle school science showed that this framework can produce tutorial planners that foster improved student problem solving and performance (Rowe, 2013; Rowe & Lester, in press). Investigating the framework's application to UrbanSim, a training environment for counterinsurgency and stability operations, serves as a useful step toward examining its generalizability, facilitated by its integration with GIFT.

Data-Driven Tutorial Planning in UrbanSim with GIFT

The proposed research on a modular reinforcement learning framework for data-driven tutorial planning will be carried out in three phases:

Collect a rich corpus of tutorial planning data from a virtual training environment for counterinsurgency and stability operations. We will conduct a series of studies in which participants—including both simulated learners and human learners—receive training for counterinsurgency and stability operations in UrbanSim. By leveraging GIFT, the training environment will be instrumented to collect data on participants' responses to scaffolding decisions, including hints, prompts, feedback, and new scenarios, that operate at both domain-dependent and domain-independent levels. Scaffolding will include both macro- and micro-adaptations through integration with the GIFT Engine for Managing Adaptive Pedagogy (EMAP). Assessments administered before, during and after the sessions will measure learning gains and problem-solving transfer, and corresponding instruments will gauge affective outcomes of motivation, self-efficacy, and situational interest.

Develop, integrate, and iteratively refine data-driven tutorial planning models for the counterinsurgency virtual training environment. By modeling the task as a collection of Markov decision processes, we will induce tutorial planning models for the virtual training environment using modular reinforcement learning. The tutorial planner will be trained using data from both simulated and human students' interaction and outcome data. The tutorial planners will control scaffolding decisions according to a probabilistic mapping between learning environment states and tutorial actions that optimizes students' learning outcomes. The tutorial planner will be integrated with GIFT via the Pedagogical and Domain Modules, and be iteratively refined over the course of the research program.

Empirically investigate the impact of data-driven tutorial planning models for the counterinsurgency virtual training environment. The research program will culminate with a study that compares the effectiveness of the final MDP-based tutorial planner to several baselines: an MDP-based tutorial planner induced with simulated student data (rather than human student data), as well as a no-planner control. Learning outcomes will be measured in terms of in-game performance, pre-to-post learning gains, and near transfer, while affective outcomes will be measured in terms of motivation, self-efficacy, and situational interest.

During the project, we will define a collection of AESs that encode a range of macro-adaptive and micro-adaptive pedagogical decisions. Once the AESs have been defined, they will be integrated with UrbanSim via GIFT, enabling an iterative design, development, evaluation, and refinement process. Initial versions of the tutorial planning policies will be induced from data generated by simulated learners, and successive versions will be defined based upon data from actual human learners. Furthermore, because AESs are encoded computationally as MDPs, we will devise tailored state representations and reward functions for each AES, which will drive the induction of tutorial policies using reinforcement learning, and be the subject of iterative refinement over the course of the project.

EXTENDING GIFT'S CAPABILITIES

A major thrust of the research program is extending GIFT's capabilities to support the creation of data-driven tutorial planners. To achieve this objective, the NCSU, IAI, and ARL teams will work collaboratively to make several extensions to GIFT.

Extended Logging Capabilities for Capturing Simulation State

In order to maintain information-rich state representations that drive run-time decisions about instructional tactics, GIFT should have access to fine-grained logs of simulation state and learner actions in the training environment. These data streams should be made available to GIFT's Pedagogical Module in order to guide decisions about scaffolding, as well as construct well-designed coaching messages—such as hints, feedback, and prompts—that are grounded with concrete details from the simulation. Competence estimates, currently provided by GIFT's Learner Module, are likely to be necessary but not sufficient for driving effective tutorial policies that enhance trainee performance and learning. Tutorial policies for individual AESs stand to benefit from granular access to tailored state representations that capture the state of the simulation environment, the history of the learner's actions, and details of the learner's state and individual traits.

Stochastic Control of Pedagogical Policies and Tactics

Throughout the project, we will deploy versions of UrbanSim, integrated with GIFT, that perform instructional interventions stochastically. MDP policies specify a probability mass function for the action choices in each state, specifying the likelihood that a specific action will be performed in a particular state according to the policy. The ability to assign probabilities to state-action pairs is critical to reinforcement learning. It provides a mechanism for stochastically exploring the space of state-action mappings, with the aim of finding an optimal policy that maximizes reward. In our framework, when we gather data from human learners using initial versions of the tutorial planner and training environment, the learners interact with a version of the tutorial planner that makes pedagogical decisions randomly. This has the effect of broadly sampling the policy space, providing valuable data for estimating the state transition dynamics of the associated MDPs, and by extension, the MDP solution policies that are induced from the corpus. In current versions of GIFT, pedagogical rules are defined symbolically, and they are deterministic. Incorporating support for probabilistic representations will be important for enabling data-driven approaches to tutorial planning that account for uncertainty in tutorial decisions and student responses.

Modular Tutorial Planning and Arbitration Procedures

The tutorial planning framework models different types of scaffolding separately, encoding each adaptable event sequence as a distinct MDP. Solution policies for AESs are also separate, and operate concurrently, interleaving decisions across multiple types of scaffolding. This enables control of a broad range of scaffolding decisions, while reducing computational challenges of tractability and data sparsity that arise for planners modeled as a single monolithic MDP (Rowe, 2013). However, by representing tutorial planning modularly, we introduce opportunities for conflicts among multiple competing policies. For example, consider a scenario involving a training environment with two simultaneously active AESs:

a *Provide-Hint* AES and an *Embedded-Assessment* AES. Suppose that the two associated policies simultaneously recommend that a hint and embedded assessment should be delivered on the same turn, but we only wish to perform a single pedagogical intervention. The two policies are in conflict; the tutorial planner has two competing action recommendations, but must choose a single course of action to follow. In situations such as this one, an external arbitration procedure is used to determine how to resolve the conflict, and thus which course of action should be taken. Arbitration procedures are typically specified manually, and can be domain-dependent or domain-independent. In our work, we often use greatest-mass arbitration, a domain-independent arbitration procedure that utilizes Q-values generated during reinforcement learning to select the course of action with greatest expected reward (Rowe & Lester, in press). To support this type of modular structure, GIFT will need to be extended to support the representation, induction, operation, refinement, and arbitration of multiple concurrent tutorial policies, which collectively control the pedagogical strategies and tactics utilized to scaffold learning.

Reinforcement Learning with GIFT

In order to increase capacity for inducing data-driven tutorial planners that are effective for a range of training environments, the proposed research program will extend GIFT by adding support for MDP representations of tutorial planning. Further, we intend to devise tools and documentation that will enable other GIFT users to automatically induce scaffolding policies using modular reinforcement learning. This will include the addition of tools to enable configurable state representations, action sets, reward functions, transition models, and learning algorithms, thus supporting implementations that are tailored to individual learning environments, but leverage reusable tools and methods from our modular reinforcement learning framework. MDP-based tutorial planning features will be integrated with GIFT via the Pedagogical and Domain Modules, and be iteratively refined over the course of the research project. These extensions will significantly streamline future efforts to apply MDP-based tutorial planning methods in particular, as well as reinforcement learning techniques in general, to new training environments and ITSs using GIFT.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

Modular reinforcement learning shows considerable promise for inducing data-driven tutorial planners for virtual training environments. We have described a research collaboration between North Carolina State University, Intelligent Automation, Inc., and the U.S. Army Research Laboratory that will investigate a modular reinforcement learning framework for tutorial planning in simulation-based virtual training environments with GIFT. The framework involves decomposing tutorial planning into multiple concurrent sub-problems, which are abstracted as adaptable event sequences (AESs). AESs are modeled as Markov decision processes, with rewards based on trainees' learning outcomes, and solution policies induced using model-based reinforcement learning. Training data for reinforcement learning is gathered from simulated and human learners' interactions with a virtual training environment, yielding policies to control a range of macro-adaptive and micro-adaptive scaffolding decisions. We intend to iteratively induce and refine data-driven tutorial planners for UrbanSim, a simulation-based virtual training environment for counterinsurgency and stability operations, over the course of the project, which will culminate with a randomized experiment that examines the induced planners' effectiveness relative to several comparison and control conditions. To enable this line of research, several extensions to GIFT will be necessary: extension of GIFT's logging capabilities for capturing simulation state and learners' action history, stochastic control of pedagogical strategies and tactics, support for modular tutorial planning and arbitration procedures, and software tools for reinforcement learning with GIFT. Results from this research will address the application of MDPs for tutorial planning across a range of learning environments, domains, student populations, and forms of coaching and scaffolding. Furthermore, the resultant improvements to GIFT will help the research community to carry out future research on automated data-driven techniques for tutorial planning.

REFERENCES

Barnes, T., & Stamper, J. (2008). Toward automatic hint generation for logic proof tutoring using historical student data. In *Proceedings of the 9th International Conference on Intelligent Tutoring Systems*, Montreal, Canada. 373–382.

Beck, J. E., Woolf, B. P., & Beal, C. R. (2000). ADVISOR: A machine learning architecture for intelligent tutor construction. In *Proceedings of the 17th National Conference on Artificial Intelligence*, Austin, TX. 552–557.

Bhat, S., Isbell, C. L., Mateas, M. (2006). On the Difficulty of Modular Reinforcement Learning for Real-World Partial Programming. In *Proceedings of the 21st National Conference on Artificial Intelligence*, Boston, MA. 318–323.

Brunskill, E., & Russell, S. (2011). Partially Observable Sequential Decision Making for Problem Selection in an Intelligent Tutoring System. In *Proceedings of the 4th International Conference on Educational Data Mining*, Eindhoven, the Netherlands. 327—328.

Chi, M., VanLehn, K., & Litman, D. (2010). Do micro-level tutorial decisions matter: Applying reinforcement learning to induce pedagogical tutorial tactics. In *Proceedings of the 10th International Conference on Intelligent Tutoring Systems*, Pittsburgh, PA. 224-234.

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.

Karlsson, J. (1997). Learning to solve multiple goals. Ph. D. Thesis, University of Rochester Department of Computer Science, Rochester, NY.

Lesgold, A., & Nahemow, M. (2001). Tools to Assist Learning by Doing: Achieving and Assessing Efficient Technology for Learning. In D. Klahr & S. Carver (Eds.), *Cognition and Instruction: Twenty-five Years of Progress*. (pp. 307-346) Mahwah, NJ: Erlbaum.

McAlinden, R., Gordon, A. S., Lane, H. C., & Pynadath, D. (2009). UrbanSim: A Game-based Simulation for Counterinsurgency and Stability-focused Operations. In *Proceedings of the AIED '09 Workshop on Intelligent Educational Games*, Brighton, UK. 41–50.

Murray, T. (2003). An Overview of Intelligent Tutoring System Authoring Tools: Updated analysis of the state of the art. In T. Murray, S. Blessing, and S. Ainsworth (Eds.), *Authoring Tools for Advanced Technology Learning Environments* (pp. 493–546). Dordrecht, the Netherlands: Kluwer.

Pokorny, B., Haynes, J., & Gott, S. (2010). Performance Assessment in Complex Environments. In Proceedings of the *Interservice/Industry Training, Simulation and Education Conference*, Orlando, Florida, December 2010.

Rowe, J. P. (2013). Narrative-centered tutorial planning with concurrent Markov decision processes. Ph.D. Thesis, North Carolina State University Department of Computer Science, Raleigh, NC.

Rowe, J., & Lester, J. (in press). Improving student problem solving in narrative-centered learning environments: A modular reinforcement learning framework. To appear in *Proceedings of the 17th International Conference on Artificial Intelligence in Education*, Madrid, Spain.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.

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

Woolf, B. P. (2008). *Building Intelligent Interactive Tutors: Student-centered strategies for revolutionizing e-learning* (p. 480). 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. 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 a 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).*

Dr. Bob Pokorny *is Director of 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. Wilbur Peng *received his B.S. in Electrical Engineering from Cornell University, Ithaca N.Y. in 1992, and received his PhD in Electrical Engineering from the University of Maryland at College Park in 2002. His dissertation research addressed the problem of representing and using expert knowledge the form of graph-based similarity structures. At IAI, Dr. Peng leads research and development projects in several areas, including cluster* *and distributed computing; autonomous aerial vehicle software control architecture; parallel discrete event simulation on multicore distributed computing clusters; network simulation, emulation and modeling; and semantic and ontological modeling methodologies, frameworks and applications.*

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.*

THEME IV:
DOMAIN MODELING

Dimensions and Challenges in Domain Modeling for Adaptive Training

Robert A. Sottilare
US Army Research Laboratory

INTRODUCTION

Today, the majority of intelligent tutoring systems (ITSs), adaptive training tools to support one-to-one and one-to-many computer-based instruction, support training and education in well-defined domains like mathematics, physics, and software programming. Since Soldiers operate in more complex, dynamic and ill-defined domains, it is necessary to expand the scope of adaptive training tools and methods to support training and education in these militarily-relevant domains. Domain modeling is a representation of knowledge for a particular task or concepts and includes: domain content (a library of scenarios or problem sets or knowledge components); an expert or ideal student model with measures of success and a library of common misconceptions; and a library of tactics or actions (e.g., questions, assessments, prompts, and pumps) which can be used by the tutor to engage or motivate the learner and optimize learning.

The adaptive training research program at the US Army Research Laboratory includes six interdependent research areas or vectors: individual learner and unit modeling, instructional management principles, domain modeling, authoring tools and methods, evaluation tools and methods, and architectural and ontological support for adaptive training. This paper examines the dimensions of domain modeling and discusses challenges in efficiently authoring domain model elements.

DIMENSIONS OF DOMAIN MODELING

The goal of our domain modeling research is twofold: (1) understand and model the characteristics, similarities, and differences of a variety of Army training domains (cognitive, affective, psychomotor, social, and hybrid) with respect to their associated knowledge representations; and (2) understand and model the dimensions (definition, complexity, and dynamics) of training domain representations in order to extend the capabilities of traditional ITSs and thereby support militarily-relevant adaptive training domains.

There are four typical elements which compose ITSs, a prime example of an adaptive training and education system: a learner or trainee model, an instructional or pedagogical model, a domain model, and some type of user interface. The domain model typically includes an expert or ideal student model by which the adaptive system measures/compares/contrasts the progress of the learner toward learning objectives. The domain model also includes the training environment, the training task and all of the associated instructional actions (e.g., feedback, questions, hints, pumps, and prompts) which could possibly be delivered by the adaptive system for that particular training domain.

One method to describe or model the domain is by the type of task which is being trained. A traditional methodology of categorizing tasks is Bloom and Krathwohl’s (1956) Taxonomy, which describes hierarchically ordered skills (competency) in the cognitive task domain. Due to the contributions of others, Bloom and Krathwohl’s original taxonomy has evolved over time to include an affective (Krathwohl, Bloom & Masia, 1964; Anderson and Krathwohl, 2001), psychomotor (Simpson, 1972) and a social hierarchy (Soller, 2001; Sottilare, Holden, Brawner, & Goldberg, 2011). The following sections describe each task domain and its relationship to adaptive training and education.

Cognitive Domain

Sometimes called the *thinking* domain, tasks in this domain stress the learner’s thinking capacity (working memory, executive control, workload management, multitasking), problem-solving and planning, decision-making, comprehension, reasoning, and attentional focus or engagement. The determination of cognitive skills may be based on learner behaviors to indicate increases in complex and abstract mental capabilities (Anderson and Krathwohl, 2001).

A revision of Bloom’s taxonomy (Anderson and Krathwohl, 2001) tracks a series of behaviors from low to high cognitive skills as follows: (1) *remembering*- the learner’s ability to recognize and recall information, (2) *understanding (also known as comprehension)* – the learner’s ability to organize, compare, and interpret information, (3) *applying*- the learner’s ability to use information to solve problems, (4) *analyzing*- the learner’s ability to examine information and make inferences from that information, (5) *evaluating* - the learner’s ability to use information to make optimal judgments, and (6) *creating*- learner’s ability to build new models (e.g., plans) from information.

Most of the ITSs in existence today focus on the cognitive task domain (Anderson, Corbett, Koedinger & Pelletier, 1995; Ritter, Anderson, Koedinger & Corbett, 2003; Graesser, Conley & Olney, 2012). Examples include model-tracing (also called example tracing) tutors which use a set of steps to walk the learner through the process of solving a problem (Koedinger, Corbett, and Perfetti, 2012). Mathematics, physics, and software programming are the most common types of model-tracing tutors. These domains constitute simple procedural tasks and are usually rule-based.

Matthews (2014) notes organizations generally do a good job of training relatively simple skills. However, a more challenging goal is to teach higher order cognitive skills such as decision-making and judgment. The Army has large investments in partial-task and scenario-based training systems which use relatively fixed strategies to guide the learner based primarily on individual and team performance measures. A concern with these systems is that soldiers learn how to win within the constraints of the system but the effect on retention and transfer is not well understood. Research is needed to build adaptivity into these training systems and thereby optimize deep learning. A goal of this research is to reduce the time to competency to allow time for automaticity through over-training and deeper learning experiences that transfer to the operational environment.

Affective Domain

Sometimes called the *feeling* domain, tasks in this domain are intended to develop *emotional intelligence* or skills in self-awareness and growth in attitudes, emotion, and feelings. The goal is to manage emotions in positive ways to relieve stress, communicate effectively, empathize with others, overcome challenges, and defuse conflict (Goleman, 2006). While listed as separate domain, affect has an interdependent relationship with cognition and learning. Specifically, confusion, frustration, boredom, surprise, delight, flow, and anxiety are considered major moderators of learning (D'Mello, 2013; Pekrun, 2006). Cognitive readiness, the capability to maintain performance and mental well-being in complex, dynamic, unpredictable environments may elicit affective responses. Dimensions of cognitive readiness, according to Kluge and Burkolter (2013), include concepts such as risk taking behavior, emotional stability and coping which may be considered part of the affective domain.

A revision of Bloom's taxonomy (Anderson and Krathwohl, 2001) tracks a series of behaviors from low-affective state to high as follows: (1) *receiving*- the learner takes in information,(2) *responding*- the learner takes in information and responds/reacts, (3) *valuing*- the learner attaches value to information, (4) *organizing* – the learner sorts information and builds mental models, *and* (5) *characterizing*- the learner matches mental models to values and beliefs ultimately influencing (e.g., promoting or limiting) the learner's behavior.

Very little training (outside of classroom-based training) is currently provided to exercise/grow skills in this important task domain and almost no adaptive training has been created to support this domain. However, D'Mello and Graesser (2012) have produced an affect-sensitive tutor that exercises emotional intelligence using the AutoTutor authoring tools. Research is needed to understand measures for the affective task domain and any unique characteristics required to author affective domain scenarios.

Psychomotor Domain

Sometimes called the *doing* or *action* domain, tasks in this domain are associated with physical tasks (e.g., marksmanship) or manipulation of a tangible interface (e.g., remotely piloting a vehicle), which may include physical movement, coordination, and the use of the motor-skills. Development of motor-skills requires practice and is measured in terms of speed, precision, distance, procedures, or techniques during execution (Simpson, 1972). Simpson's hierarchy of psychomotor learning ranges from low to high: *perception*–the ability to use sensory cues to guide motor activity; (1) *set or readiness to act*; (2) *response*–early stages of learning a complex skill through imitation and trial and error; (3) *mechanism*–habitual learned responses; (4) *complex overt response*–skillful performance of complex movements; (5) *adaptation*–well-developed skills that are modified to support special requirements; and (6) *origination*–the development of new movement patterns to fit unique situations.

While this domain is well represented in Army training, research is needed to build adaptiveness into these training systems and thereby optimize deep learning. Again, a goal of this research is to reduce the time to competency to allow time for automaticity through over-training and deeper learning experiences that transfer to the operational environment.

Social Domain

Sometimes called the *collaborative* domain, tasks in this domain include a set of collaborative characteristics or measures of learning in the social domain as defined by Soller (2001): (1) *participation*, (2) *social grounding*- team members "take turns questioning, clarifying and rewording their peers' comments to ensure their own understanding of the team's interpretation of the problem and the proposed solutions", (3) *active learning conversation skills* - quality communication, (4) *performance analysis and group processing* - groups discuss their progress, and decide what behaviors to continue or change (Johnson, Johnson, & Holubec, 1990) and (5) *promotive interaction* - also known as win-win, this characteristic occurs when members of a group perceive that they can only attain their goals if their team members also attain their goals.

The Program for International Student Assessment (PISA, 2015) is a worldwide study by the Organization for Economic Cooperation and Development (OECD) in both member and non-member nations. This study focuses on 15-year-old students and their scholastic performance in mathematics, science, and reading. During PISA 2015, OECD defined a matrix of collaborative problem solving skills. Some of the skills and associated behaviors in this matrix may also apply to situational problem solving in the Army (e.g., staff level organizations evaluating options to meet objectives during military operations). Research is needed to determine if the model in this matrix will generalize beyond its original application to 15 year old students.

Research is needed to create and apply measures for both collaborative learning and team training activities. The interdependent nature of Army tasks also requires tutoring of squads and other echelons of teams (collective training). Research is also needed to develop team state models to drive adaptive training decisions. An extensive review of the team performance and tutoring literature has been conducted (Burke, Sottilare, Salas, Johnston, Sinatra, & Holden, 2015, *in press*) to determine antecedents of team outcomes (performance, learning, satisfaction, and viability) which include behavioral measures along with models of cooperation and team cognition. Additional research in the social domain may be found on team cognition (Salas & Fiore, 2004), team mental models (Fletcher & Sottilare, 2013; Rouse, Cannon-Bowers & Salas, 1992), and situational awareness in team performance (Salas, Prince, Baker & Shrestha, 1995).

DOMAIN MODELING GOALS AND CHALLENGES

A foundational goal of adaptive training research at the U.S. Army Research Laboratory (ARL) is to *model the perception, judgment, and behaviors of expert human tutors* to support practical, effective, and affordable learning experiences guided by computer-based agents. To this end, five primary goals for domain modeling within adaptive training systems have been identified and are discussed in this section along with the major challenges or barriers to success.

Representing Domain Attributes

Our first goal is to conduct research to determine the influence of task domain attributes (e.g., complexity, definition, and dynamics) on cognitive mechanisms (e.g., learning, comprehension, performance, retention, reasoning, and transfer of knowledge and acquired skills to the operational environment). Research is needed to analyze related and unrelated domains to determine commonalities, performance measures, and user requirements.

The challenge in meeting this goal is since each domain contains unique domain knowledge, the relationships of these attributes to the cognitive mechanisms listed previously are not well understood across various task domains (cognitive, affective, psychomotor, social, and hybrid tasks). It may be possible to generalize relationships in a particular task domain with the goal being to identify attributes with high effect on cognitive mechanisms.

Also important to this research goal is the ability to define how these attributes are measured, how qualitative inputs are going to be assessed against quantitative metrics, and how stakeholder requirements and learner-generated content (e.g., social media input on relevance and impact) influence the modeling of each domain. There needs to be a match between the desired requirements and what is actually authored in the domain. For example, tutors for marksmanship for the Army and Marines may be very similar based on the task domain (psychomotor) but may also have different instructional methods or concepts based on organizational requirements.

Reducing Time and Skill to Author Domain Knowledge

Our second goal is to discover and improve authoring tools and methods to more easily represent domain knowledge including methods to select optimal instructional tactics (actions) based on sound instructional strategies (plans), the instructional context, and the learner's states and traits. Specifically, our goal is to provide sufficient domain knowledge to effectively adapt the training of the task for individual learners and units while significantly lowering the cost and skills needed to author adaptive training systems for the U.S. Army.

Authoring in complex domains is a time intensive process and research is needed to determine what attributes of the domain model influence successful outcomes (e.g., skill development) and antecedents to those outcomes (e.g., motivation, engagement, and grit) which may influence the complexity, definition, and dynamics of the domain knowledge (e.g., content, feedback, and assessments) presented to the learner and thereby influence the cost.

Adaptive training systems by their nature offer more flexibility and are tailored to individual learners. Given the variability of learner attributes across the general population, this creates a greater demand for domain authoring. Finding efficient methods to create new content and to reuse existing content (e.g., training content in existing Army training simulations) should be a priority for domain modeling research. Specifically, we need to examine methods to automate large portions of the authoring process including the automated development of expert models (sometimes called ideal student models), the automated generation of scenario variants from base cases, and the authoring of assessments from which the ITS determines progress and corresponding strategies and tactics.

Improving the Interoperability of Domain Models

Our third goal is to develop standards for interoperability to promote reuse of domain knowledge (e.g., content, expert models, question banks, assessments, and tactics). Specifically, our goal is to set interoperability standards for U.S. Army training resulting in reduced development and maintenance costs and increased speed of adoption for adaptive training technologies. The major challenges are: no standards exist for adaptive training; and interoperability, the ability to pull and replace one model or domain element with another authored elsewhere, is extremely low. Effort should be expended to work with the ITS community to develop interoperability standards to maximize reuse of domain knowledge. Defining standard methods of representing and interfacing with domain knowledge will allow a single adaptive training architecture or framework to support multiple domains and reduce authoring costs. This is the primary driver for the development of the Generalized Intelligent Framework for Tutoring (GIFT).

The U.S. Army's focus on scenario-based training along with the significant investment in training infrastructure should also be considered in developing interoperability. To this end, GIFT includes a standard interface specification, or gateway, so that Army training systems that meet this specification can be integrated to provide adaptive training capabilities. In addition, the GIFT gateway currently has an IEEE 1278 Distributed Interactive Simulation (DIS) standard interface to receive tactical data (e.g., entity location) to support instructional decisions and to push instructional tactics (e.g., interaction with the learner or changes to the training environment) to DIS compatible training simulations.

Optimizing the Selection of Tactics

Our fourth goal is to optimize the selection of tactics, domain-specific actions by the tutor, in order to provide the greatest opportunity for performance, learning, retention, and transfer. In GIFT, tactics are the actions taken by the tutor in response to learner states and instructional context (e.g., conditions of the scenario or problem presented) as shown in Figure and are constrained by available options provided during the authoring process. Improving the usability and efficiency of the authoring tools will likely result in a greater number of available options for adaptive training domains.

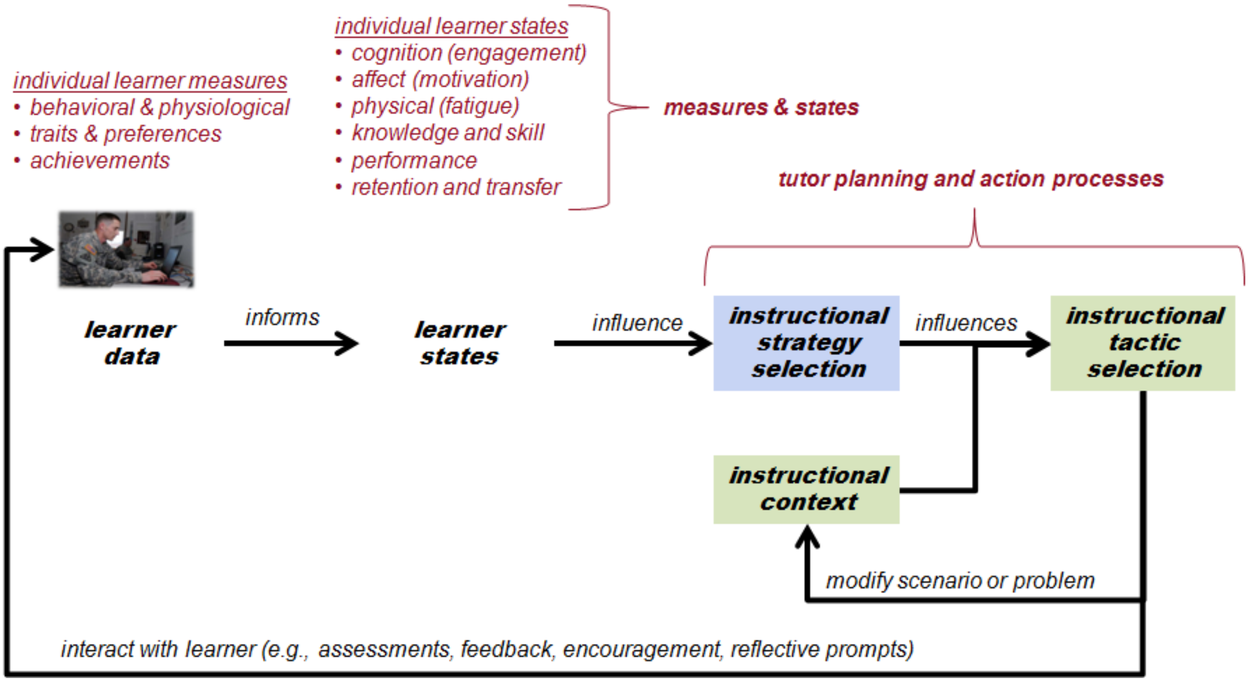

individual learner states
• cognition (engagement)
• affect (motivation)
• physical (fatigue)
• knowledge and skill
• performance
• retention and transfer
measures & states
individual learner measures
• behavioral & physiological
• traits & preferences
• achievements
tutor planning and action processes
learner data
informs
learner states
influence
instructional strategy selection
influences
instructional tactic selection
instructional context
modify scenario or problem
interact with learner (e.g., assessments, feedback, encouragement, reflective prompts)

Figure 1. Updated Individual Learning Effect Model in GIFT

Unlike instructional strategies, which are derived from good pedagogical practices based learning theory and influenced by the learner's states, tactics are domain-specific actions by the tutor and may not be generalized across all task domains. Research is needed to determine methods to select the best possible tactic given the selected instructional strategy, the training domain, and the availability of tactics.

Modeling the expert behaviors of human tutors may be a starting point, but accurate assessment methods are needed for both individual and team level states. These states are critical in selecting appropriate strategies (plans for action) and tactics (actions – e.g., assessments, feedback, questions, changes to the training environment) per the Learning Effect Model (Sottilare, 2012; Fletcher & Sottilare, 2013; Sottilare, 2013; Sottilare, Ragusa, Hoffman & Goldberg, 2013) as updated for both individuals (Figure) and teams (Figure). Assessment of team states may also be useful in determining constraints to be monitored by tutoring agents and interactions between the learner and the training environment.

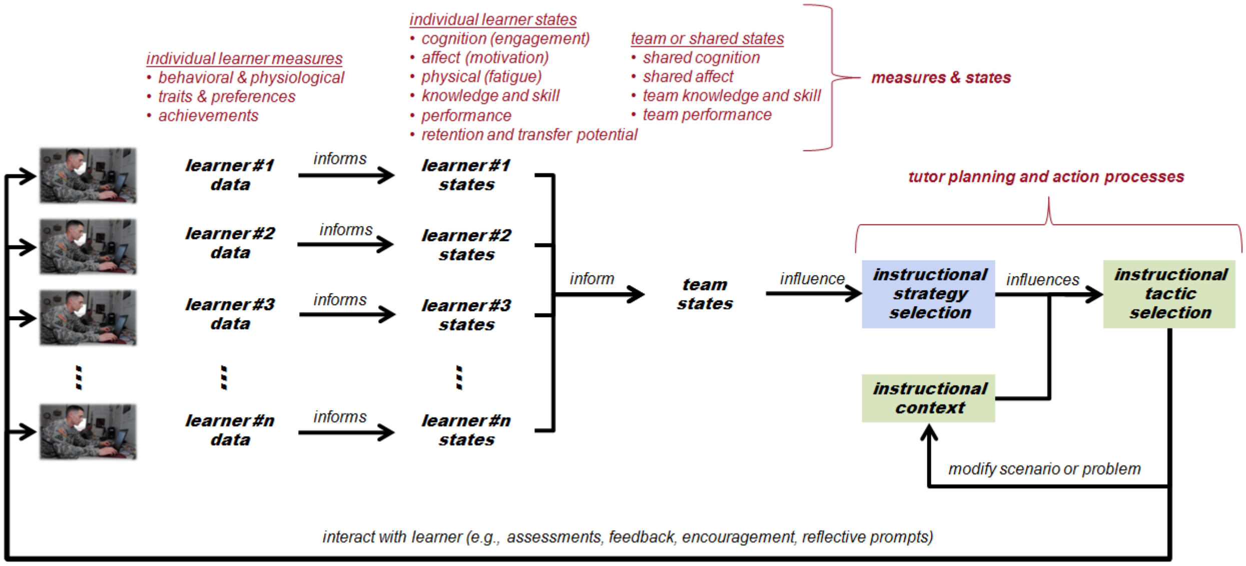


Figure 2. Updated Team Learning Effect Model in GIFT

Extending Adaptive Training to Militarily-Relevant Domains

Our fifth goal is to be able to extend adaptive training to support militarily-relevant domains. Many military tasks are hybrids of task domains in that they include aspects of cognitive (thinking – evaluating, problem-solving, and decision-making), affective (feeling – making value judgments), psychomotor (doing – physical action), and/or social (collaborating – working in teams). Army training differs greatly from traditional ITSs which are primarily problem-based (e.g., mathematics, physics, computer programming) and generally vary only in complexity. Given much of Army training is scenario-based, the *realism* of the training environment, *accessibility* of the training, the *complexity* of the scenario, the *physical dynamics* of the task, and the variable level of *definition* are all design considerations for adaptive training systems for military use. It will be essential to match the attributes of the environment to the task domain by asking the question “what is necessary to train the task effectively”. This variability in adaptive training and educational domains will allow for greater opportunities for Soldiers to train at the point-of-need and to train more closely to how they fight. This is anticipated to result in greater learning, performance, retention, and transfer of skills to the operational environment.

As in all training, it will be essential to match the realism of the training environment (e.g., serious game, virtual simulation, embedded training) to enable learners to progress toward established learning objectives. For embedded training where Soldiers bring training with them in their operational platform (ground, air, sea, dismounted), considerations should be made regarding the visual resolution of virtual elements of the training environment and what is required to train the task (Sottilare, Marshall, Martin, & Morgan, 2007). For example, if the resolution of virtual targets is insufficient to support either detection or identification of the targets at comparable distances to the real world (also known as live environment) then negative training may result.

Another consideration in militarily-relevant domains is accessibility. Accessible learning is being directly addressed by the research and development of GIFT (Sottilare et al., 2012; Sottilare, et al, 2013), an adaptive tutoring architecture that is modular and service-oriented. GIFT will also support access to adaptive tutoring resources (domain content, assessments, Web services) via the Internet and allow content to be presented in a Web browser.

Adaptive training solutions must be able to include the complexities of each to provide tailored training across the broad spectrum of Soldier tasks. This includes the ability to align more closely with the nature of those tasks in order to promote transfer of skills from training systems to the theaters of operation. Ultimately, this will mean moving from desktop training environments to more interactive and physical environments. Research will be needed to examine a learning progression from the *desktop* to the *wild*, a concept where soldiers can received training anywhere they happen to be. We examine four modes of adaptive training environments to support this concept.

Variable task dynamics refer to the physical modes of interaction of the learner during the training experience. This ranges from static (seated position for desktop training) to limited dynamic (standing position limited range of mobility in instrumented areas) to enhanced dynamic (standing, kneeling, and prone positions with expanded mobility in instrumented areas) to "in-the-wild" (any position with unlimited mobility where sensors and communications move with the Soldier).

Variable task definition refers to how well the domains are understood in terms of standards and measures of performance. Well-defined domains (e.g., mathematics) typically have one correct path to a successful outcome and a set of specific standards for measuring success. Ill-defined domains may have multiple paths to successful outcomes, and they tend to have vague standards and less defined measures of success. Ill-defined domains may also have unexpected and inconsistent confounds which could cause learning to be perceived when there really is a mediating underlying factor. Analyzing these can provide greater knowledge to answering the why behind performance and learning outcomes.

Finally, task complexity refers to the range of difficulty in understanding and performing the task. Task complexity can range from simple procedural tasks to more complex multidimensional tasks.

Next, we examine modes of dynamic interaction. Limited dynamic environments support hybrid (cognitive, affective, psychomotor) tasks where a larger degree of interaction with the environment and other learners is critical to learning, retention, and transfer to the operational environment. Decision-making and problem-solving tasks may be taught easily in a limited dynamic mode along with tasks requiring physical orientation (e.g., land navigation).

Enhanced dynamic environments support tasks where freedom of movement and a high degree of interaction with other learners are critical to learning, retention, and transfer to the operational environment. Building clearing and other team-based tasks may be taught easily in an enhanced dynamic mode.

In the wild mode is transferring tutoring to the operational environments and could also be called embedded training for Soldiers. In the wild mode is critical to support tasks where a very high degree freedom of movement and a high degree of interaction with other learners are critical to learning, retention, and transfer to the operational environment. It is anticipated that psychomotor and social tasks may be best taught in the wild or an environment more closely resembling the operational environment.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

GIFT is being developed to support a broader variety of domains than ITSs support today. To support adaptive training in these domains, we have identified five goals and their affiliated challenges. It will be critical to be able to represent domain attributes to support ITS authoring. It will also be important to reduce the time and skill to author domain knowledge so ITSs are a more appealing and affordable choice for training. Improving the interoperability of domain knowledge will promote reuse between domains. The actions of the tutor must be tailored to the needs of the learner and selected to optimize learning outcomes. Finally, we envision significant challenges in applying adaptive training solutions to militarily-relevant training domains which include aspects of multiple domains (e.g., land navigation tasks which involve recognition and decision-making in the cognitive domain mixed with physical aspects of moving over uneven terrain).

REFERENCES

Anderson, J. R.; Corbett, A. T.; Koedinger, K. R.; Pelletier, R. Cognitive Tutors: Lessons Learned. *The Journal of the Learning Sciences* 1995*, 4*,167–207.

Anderson, L. W.; Krathwohl, D. R. (Eds.). A Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives: Complete Edition; New York: Longman, 2001.

Bloom, B. S., & Krathwohl, D. R. (1956). Taxonomy of educational objectives: The classification of educational goals. *Handbook I: Cognitive domain*.

Burke, S., Sottilare, R., Salas, E., Johnston, J., Sinatra, A., and Holden, H. (2015, *in press*). Towards a Scientifically-Rooted Design Architecture of Team Process and Performance Modeling in Adaptive, Team-Based Intelligent Tutoring Systems: Methodology for Literature Review and Meta-Analysis, and Preliminary Results. *Army Research Laboratory* (ARL-TR-XXXX).

D'Mello, S. K. (2013). A selective meta-analysis on the relative incidence of discrete affective states during learning with technology. *Journal of Educational Psychology*, 105,1082-1099.

D'Mello, S. K. & Graesser, A. C. (2012). AutoTutor and affective AutoTutor: Learning by talking with cognitively and emotionally intelligent computers that talk back. ACM Transactions on Interactive Intelligent Systems, 2(4), 23: 1-38.

Fletcher, J.D. and Sottilare, R. (2013). Shared Mental Models and Intelligent Tutoring for Teams. In R. Sottilare, A. Graesser, X. Hu, and H. Holden (Eds.) Design Recommendations for Intelligent Tutoring Systems: Volume I - Learner Modeling. Army Research Laboratory, Orlando, Florida. ISBN 978-0-9893923-0-3.

Goleman, D. (2006). *Emotional intelligence*. Bantam.

Graesser, A. C., Conley, M. W., & Olney, A. M. (2012). Intelligent tutoring systems. In S. Graham, & K. Harris (Eds.), *APA Educational Psychology Handbook: Vol. 3. Applications to Learning and Teaching* (pp. 451-473). Washington, DC: American Psychological Association.

Johnson D, Johnson R, Holubec EJ. Circles of learning: Cooperation in the classroom (3rd ed.). Edina, MN: Interaction Book Company, 1990.

Kluge, A., & Burkolter, D. (2013). Enhancing Research on Training for Cognitive Readiness Research Issues and Experimental Designs. *Journal of Cognitive Engineering and Decision Making*, 7(1), 96-118.

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

Krathwohl, D. R.; Bloom, B. S.; Masia, B. B. *Taxonomy of Educational Objectives: Handbook II: Affective Domain*; New York: David McKay Co, 1964.

Matthews, M.D. (2014). Head Strong: How Psychology Is Revolutionizing War (p. 204). Oxford University Press, New York.

Pekrun, R. (2006). The control-value theory of achievement emotions: Assumptions, corollaries, and implications for educational research and practice. *Educational Psychology Review*, 18, 315-341.

Program for International Student Assessment (PISA, 2015). National and International Assessment Activities 2013-2014. Available from: http://nces.ed.gov/nationsreportcard/subject/about/pdf/2013_2014_nces_national_and_international_assessment_activities_schedule.pdf

Ritter, S., Anderson, J. R., Koedinger, K. R., Corbett, A. (2007) Cognitive Tutor: Applied research in mathematics education. Psychonomic Bulletin & Review, 14, 249-255

Rouse, W. B., Cannon-Bowers, J. A., & Salas, E. (1992). The role of mental models in team performance in complex systems. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(6), 1296-1308.

Salas, E. E., & Fiore, S. M. (2004). Team cognition: Understanding the factors that drive process and performance. *American Psychological Association*.

Salas, E., Prince, C., Baker, D. P., & Shrestha, L. (1995). Situation awareness in team performance: Implications for measurement and training. Human Factors: The Journal of the Human Factors and Ergonomics Society, 37(1), 123-136.

Simpson, E. The Classification of Educational Objectives in the Psychomotor Domain: The Psychomotor Domain, Vol. 3; Washington, DC: Gryphon House, 1972.

Soller, A. (2001). Supporting social interaction in an intelligent collaborative learning system. International Journal of *Artificial Intelligence in Education*, 12(1), 40-62.

Sottilare, R., Marshall, L., Martin, R. & Morgan, J. (2007). Injecting realistic human models into the optical display of a future land warrior system for embedded training purposes. *Journal for Defense Modeling and Simulation*. April 2007, Volume 4, Number 2.

Sottilare, R., Holden, H., Brawner, K., & Goldberg, B. (2011). Challenges and Emerging Concepts in the Development of Adaptive, Computer-based Tutoring Systems for Team Training. In Proceedings of the *Interservice/Industry Training Simulation & Education Conference*, Orlando, Florida, December 2011.

Sottilare, R. Considerations in the Development of an Ontology for a Generalized Intelligent Framework for Tutoring. *International Defense & Homeland Security Simulation Workshop in Proceedings of the I3M Conference*. Vienna, Austria, September 2012.

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.; Holden, H.; Goldberg, B.; Brawner, K. *The Generalized Intelligent Framework for Tutoring (GIFT)*; In C. Best, G. Galanis, J. Kerry and R. Sottilare (Eds.) Fundamental Issues in Defence Simulation & Training. Ashgate Publishing, 2013.

Sottilare, R. (2013). Special Report: Adaptive Intelligent Tutoring System (ITS) Research in Support of the Army Learning Model - Research Outline. Army Research Laboratory (ARL-SR-0284), December 2013.

Sottilare, R.; Ragusa, C.; Hoffman, M.; Goldberg, B. Characterizing an Adaptive Tutoring Learning Effect Chain for Individual and Team Tutoring. In *Proceedings of the Interservice/Industry Training Systems & Education Conference*, Orlando, Florida, December 2013.

ABOUT THE AUTHOR

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.*

The Application of GIFT in a Psychomotor Domain of Instruction: A Marksmanship Use Case

Benjamin Goldberg[1], Charles Amburn[1]
U.S. Army Research Laboratory—Human Research and Engineering Directorate[1]

INTRODUCTION

The Generalized Intelligent Framework for Tutoring (GIFT) is a domain-agnostic standard for building adaptive training functions and Intelligent Tutoring Systems (ITSs) across an array of tasks and an unlimited set of knowledge, skills, and abilities (KSAs). A current goal is to extend GIFT into Army-valued skill domains that incorporate psychomotor components of task execution. The intention is to broaden ITS modeling techniques beyond the traditional cognitive dimensions of learning that account for problem-solving and decision-making processes. The aim is to enhance training systems to support physical skill development on both an individual and team level through deliberate practice techniques (Ericsson, 2006, 2014) managed pedagogically by ITS assessment and feedback methods.

From an implementation standpoint, applying adaptive training to a psychomotor domain requires the same piece parts associated with developing any ITS; data representations at a granular enough level to inform appropriate assessments, established models of expert performance to inform performance state determinations, and a pedagogical model that guides practice and accelerates skill acquisition through formative feedback and adaptive sequencing of scenarios. For an initial use case, we identified the domain of marksmanship as an excellent candidate to guide development efforts. This is due to the Army's large investment in marksmanship dedicated simulators that provide interactive hands-on training across live range type exercises and story-driven scenarios.

Rifle Marksmanship and Adaptive Training Potential

Rifle marksmanship is a complex psychomotor skill demanding high physical and mental coordination. As with any complex skill component, training starts with the basics. Basic rifle marksmanship (BRM) involves the execution of fundamental procedures to consistently strike a target in a manner that can be replicated over multiple trials. The Engagement Skills Trainer (EST) is a simulated firing range designed as a cost-saving solution for deliberate practice of BRM fundamentals, and allows individuals to realistically replicate the procedures of operating a rifle for the first time in a controlled and safe environment. To effectively monitor performance and diagnose error, the weapons used in the EST are instrumented with various sensor technologies (e.g., trigger pressure sensor, accelerometer, breathing strap, and aim trace laser). These data streams integrated in the EST provide visual tools for instructors to observe trainee behavior leading up to and following the execution of a shot so as to better assess performance and diagnose issues (see Figure 1). This puts the responsibility of diagnosis and remediation directly on the instructor, resulting in subjective opinions driving training practices.

There are a couple recognized issues to this approach. First, prior research has shown a lack of consistency between BRM instructors in terms of error diagnosis (James & Dyer, 2011), and often, the data itself is overlooked due to difficulties associated with accessing and efficiently interpreting data outputs. Secondly, the throughput of trainees on the EST for BRM related training is quite large, making it relatively impossible to provide personalized instruction for all. Currently it is common to have 15 trainees at a time on an EST with only 1-2 instructors providing support.

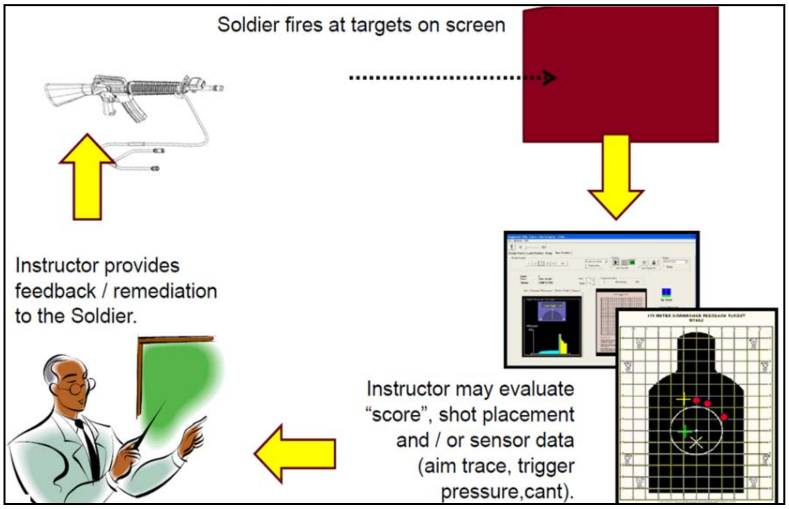

Soldier fires at targets on screen
Instructor provides feedback / remediation to the Soldier.
Instructor may evaluate "score", shot placement and / or sensor data (aim trace, trigger pressure,cant).

Figure 1. Current BRM Training in the EST

In an effort to enhance the EST to support tailored instruction across all users, work has started on integrating adaptive training technologies that enable real-time performance diagnosis for triggering objective-based guidance and remediation (Amburn, Goldberg, & Brawner, 2014; Goldberg, Amburn, Brawner, & Westphal, 2014). We are investigating how GIFT can be applied to consume EST sensor data for the purpose of constructing data-driven assessment logic. The goal is to establish robust modeling techniques that classify sensor stream inputs against a set of designated behavioral objectives that align with BRM fundamentals. In this instance, GIFT would consume both performance derived outcomes calculated within the EST and raw sensor data logged during task execution. Based on designated domain representations, GIFT would analyze inputs against a set of criteria to gauge performance and diagnose error if appropriate. GIFT could then apply pedagogical reasoning and direct feedback to the trainee through a communication interface; or data and feedback could be communicated directly to the instructor to support a data-informed human intervention (as seen in Figure 2).

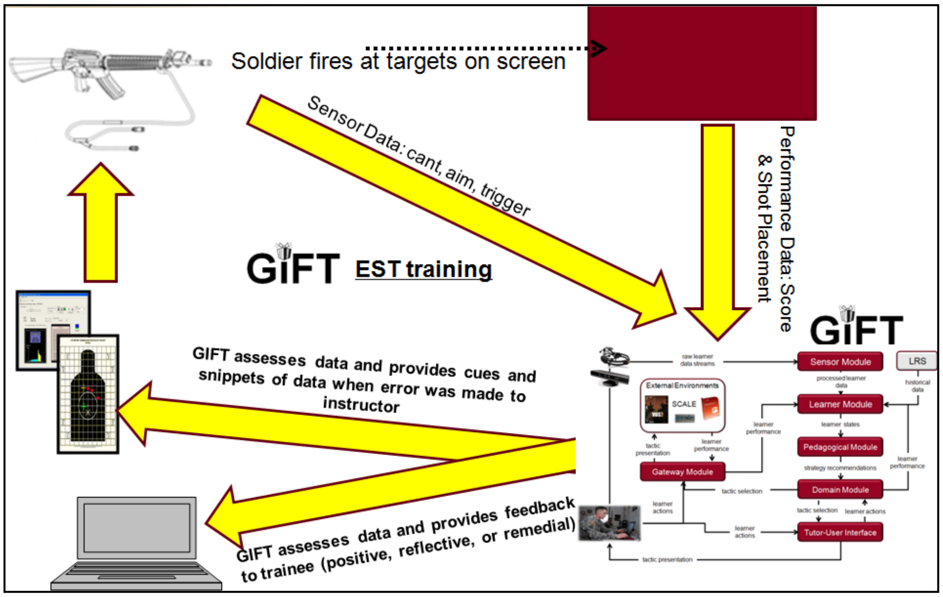


Figure 2. BRM Training Using the EST and GIFT

In this paper, we will present development efforts in GIFT to support a BRM use case. We will also discuss dependencies and limitations of modeling techniques applied, along with other recognized techniques being considered for future application. In addition, we are interested in feedback practices; specifically, what feedback techniques are best when managing a psychomotor-based procedural task and what communication modalities are most effective at relaying feedback that will optimize performance outcomes. Finally, we will present an experimental design to inform our first study, followed by themes of future research that will potentially be pursued within this domain of instruction.

GIFT DEVELOPMENT EFFORTS

To support a psychomotor use case of adaptive marksmanship, there are a number of development efforts required within the GIFT architecture (see Figure 3). These tasks include: establishing communication between the training environment and GIFT through an EST interop plug-in in the gateway module, building a set of domain models that manages assessment practices, configuring a learner model to inform state representations of performance and behavior, configuring a pedagogical model to manage feedback and remediation functions, and selecting a communication modality by which to present information back to the learner. In this section, we provide a high-level breakdown of each task and the considerations associated with design implementations.

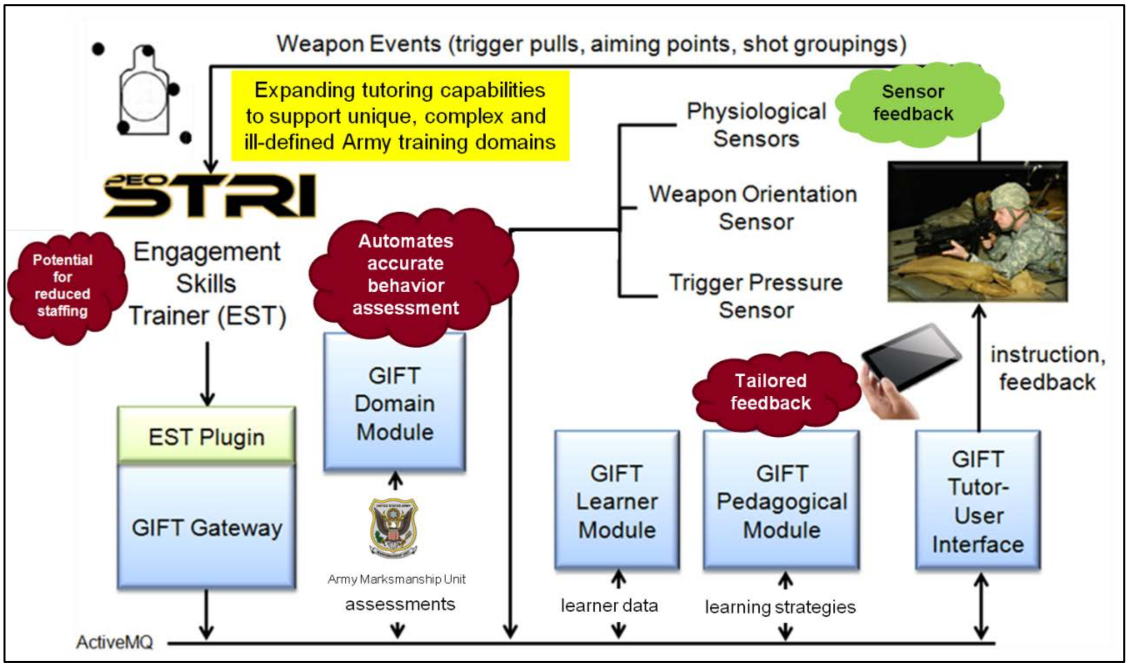

Weapon Events (trigger pulls, aiming points, shot groupings)
Expanding tutoring capabilities to support unique, complex and ill-defined Army training domains
PEO STRI
Potential for reduced staffing
Engagement Skills Trainer (EST)
Automates accurate behavior assessment
Physiological Sensors
Sensor feedback
Weapon Orientation Sensor
Trigger Pressure Sensor
instruction, feedback
Tailored feedback
EST Plugin
GIFT Gateway
GIFT Domain Module
Army Marksmanship Unit
assessments
GIFT Learner Module
learner data
GIFT Pedagogical Module
learning strategies
GIFT Tutor-User Interface
ActiveMQ

Figure 3. GIFT Architecture in Support of Adaptive Rifle Marksmanship

Data Capture through GIFT's Gateway

The first task is to build a training environment plug-in in GIFT's gateway module to establish a data exchange capability. This allows GIFT to receive and send data to and from the marksmanship simulator. For this specific application, GIFT was configured to receive performance data informed by the system's scoring algorithms, along with raw data produced by sensors embedded directly on the weapon. To further enhance GIFT's logging capability for a psychomotor use case, additional sensors were interfaced to provide data linked with breathing, heart rate, and arousal (measured through electrodermal activity). This integration takes place within GIFT's sensor module. It is also in the sensor module where filters can be applied to data channels that transform raw inputs into specified metrics that can be used to inform assessment practices (e.g., converting raw electrocardiogram data into a measure of heart rate variability).

In applying adaptive training practices to BRM training methods, these data fields provide the information necessary to monitor performance and diagnose error through behavior representations captured across the multiple sensors. With the completion of this task, GIFT is able to capture and log these data fields with synchronized timestamps. This synchronization is used to build data sets for model development, as well as to manage data filtering and assessment practices. The next task is to establish models that can be applied to this data for informing assessment methods.

Assessment in GIFT's Domain Module

Building assessment logic in GIFT is done within the domain module. Conditions are written in the domain module that designate performance state information that guides GIFT's Adaptive Tutoring Learning Effect Chain (ATLEC; Sottilare, Ragusa, Hoffman, & Goldberg, 2013). To do so, first an author establishes a representation of task concepts and objectives by which a learner will be assessed against. This is traditionally informed through a domain or task analysis. In terms of BRM, the concepts and objectives selected for our use case were taken directly from U.S. Army Field Manual (FM) on Rifle Marksmanship (FM 3-22.9; Army, 2011). We selected the four 'fundamentals' of marksmanship taught in BRM. These include: (1) trigger squeeze; (2) breathing; (3) body position; and (4) aiming. These fundamentals are organized within GIFT's modeling schema standard. The schema was designed to support domain-independency across all remaining modules within the GIFT architecture. It allows an author to establish an ontological representation of concepts and objectives that highlight the level of granularity associated with its assessment.

In the context of a psychomotor use case, assessment practices are inherently linked to two factors: performance and behavior. In BRM training, the critical variable that determines qualification is performance. If a trainee can consistently meet performance standards, then they are advanced to the next period of instruction. If a trainee does not meet standard, then coaching is required. Yet coaching cannot be based solely on performance outcomes. As such, psychomotor adaptive training requires complimentary assessments that aid in identifying what aspects of the domain to focus on. As an example, if a basketball player exhibits an unconventional approach to shooting free throws as identified by some assessment logic, yet he makes a high percentage of shots, should the system intervene to correct his behavior? Or should the system recognize the superior performance level and allow the behavior to persist? Though some coaching might improve even the superior performer, it is often in the case of the low performer where immediate coaching is necessary.

In this instance, coaching is based on moderators of performance. For marksmanship, these moderators are behavior patterns exhibited during task execution. This behavior is inferred from weapon sensors and physiological markers recorded during task execution. As such, assessments in GIFT link data types with the concepts they inform. As an example, the fundamental of trigger squeeze will be directly linked to the trigger sensor embedded directly on the weapon. With this linkage, a model will be developed to inform state representations that will guide pedagogical decisions. These state representations consist of four levels: (1) unknown, (2) below-expectation, (3) at-expectation, and (4) above-expectation. A current open question is what type of model is most suitable in a psychomotor-based training domain. Two approaches we are considering include: expert models of desired performance and buggy-libraries that associate with common novice error.

Expert Model Development

The initial approach we are applying to the domain of marksmanship is building mathematical representations of expert behavior that can be used to infer novice behavior against (Amburn et al., 2014). Preliminary work on this task was conducted on a customized marksmanship at-home trainer, with results showing experts to exhibit similar behaviors when performing grouping exercises while prone and kneeling (Goldberg et al., 2014). Expert models were generated for all four BRM fundamentals, and were developed from data collected across a set of eight expert performers; in this case, the Army Marksmanship Unit's Service Rifle Team. As the subjects in this model development were the Army's best shots, the variance in performance was rather small. This resulted in the development of descriptive models that quantitatively represented behavior as a range of values within two standard deviations of a metric's absolute mean value. These approaches passed cross-fold validation practices, providing evidence that experts perform distinctively similar behaviors when conducting basic marksmanship drills (Goldberg et al., 2014).

Yet, how effective are these models at informing feedback practices within an adaptive training environment? A model of this nature can tell you what a trainee is doing that is not reflective of expert performance, but offers no insight on the specific error that individual is making. This limits the type of pedagogy a system like GIFT can provide. As such, the first round of experiments will focus on generic remediation materials that map directly to a fundamental concept being violated (i.e., if a trainee is assessed poorly on trigger squeeze, then GIFT will present feedback linked directly to proper execution of that behavior). This approach also keeps models independent of one another in terms of machine learning techniques, as the performance state space is only that of expert. A plan on the next set of data is to establish predictive regression models that designate which behaviors have the highest impact on performance scores.

Buggy Library Development

In an effort to build more informative assessment models in BRM, an approach we will apply once a testbed is established involves the development of a buggy library (i.e., model of misconceptions). In other words, we want to model common errors novices make when learning BRM procedures. This approach varies from expert models, as it takes behavior data and determines a specific type of error made among a bank of available choices. This bank is based on training doctrine, as well as opinion of expert marksmen as they observe novices perform BRM procedures. In essence, this becomes a machine learning problem. Expert annotators are instructed to observe and classify behavior based on available data streams. Common errors will include things like improper breathing, squeezing the trigger too quickly, poor body alignment, etc. This creates a rich state space to drive machine learning methods to identify patterns in data that consistently designate this behavioral outcome.

This method is prone to error as the annotation of novice mistakes is subjective by nature. Inter-rater reliability is key to this method, as mutual agreement across experts will assist in creating data-driven models that are representative of expert opinion. A potential pitfall is the recognized inconsistency of instructors in BRM training schoolhouses. James and Dyer (2011) note that trainers of BRM vary in the tactics and procedures they teach along with their complete understanding of the behaviors involved in the task (Goldberg et al., 2014). Nonetheless, we will examine the feasibility of producing a buggy library capability in GIFT to inform BRM assessment practices. This will be conducted during our first interaction with novice performers in an effort to validate the application of expert models in identifying non-expert behavior.

State Representation in GIFT's Learner Module

Once assessment practices are completed within GIFT's domain module, performance states are communicated up to the learner module for the purpose of constructing an overall representation of the trainee's state. This state representation is ultimately used to prescribe a pedagogical strategy. As an example in the domain of BRM, a subject will be assessed on five-shot groupings at a time. As such, the domain module will receive performance outcomes and classify behavior based on data across all five shots. Following, performance and behavior states will be determined and sent in a message for the learner module to act on. It is in the learner module where domain specified performance and behavior data are combined with learner affect data to produce a complete learner state. This might associate poor performance on concept 1, which is trigger squeeze, with an affective state of high arousal. This state representation displays how an individual is performing as well as how they are feeling at a given moment within the training event. Information that also gets applied to the learner state is available trait-based information that might influence pedagogical determination. This could include more persistent attributes of a learner, such as their motivation, grit, etc. With the production of a complete learner state, the learner module then communicates this message onto the pedagogical module for the purpose of moderating the selection of instructional strategies.

Instructional Management in GIFT’s Pedagogical Module

GIFT’s pedagogical module is designed to act on learner state information. It observes performance states across all concepts linked to the domain module’s ontological representation and applies selection criteria to determine what strategy requests to pass down to the domain module for execution. At the current moment, GIFT supports four types of strategy requests: (1) provide guidance, (2) perform scenario adaptation, (3) administer further assessment, and (4) do nothing. Each of these strategies is represented in a domain-agnostic fashion and associate with very generic descriptors of what to do in actuality. These strategy calls are specific to a given concept or sub-concept in GIFT’s domain module, and link directly with tactics. These tactics are domain and context specific actions that GIFT can execute for a given concept and with a given strategy request (Sottilare, Graesser, Hu, & Goldberg, 2014). As an example, the pedagogical module may send a ‘provide guidance’ strategy request for concept 2 (trigger squeeze). The domain module will receive this request and execute the tactic previously authored by the developers of the marksmanship tutor. In this instance, it may be a video that reviews proper application of trigger squeeze or it could be a subtle prompt intended to get a learner to activate prior knowledge and focus on the fundamentals of proper trigger control. In the initial testbed, tactics associated with the provided guidance strategy request will be authored for all four fundamentals being tracked.

It is worth noting that in the next release of GIFT, strategy requests will be extended to support further granularity of strategy types. This will provide an additional layer of abstraction that extends GIFT’s personalization capabilities. This may involve adding dimensions of specificity and timing to a ‘provide guidance’ strategy request, dimensions of difficulty and complexity to a ‘perform scenario adaptation’ request, and adding distinctions across assessment types through the incorporation of Bloom’s Taxonomy (Krathwohl, 2002).

In the context of psychomotor instruction, another factor to consider with relation to instructional management is communication modality (i.e., the way in which information and guidance is communicated to the learner). In the majority of GIFT use cases, communication is managed through the Tutor User Interface (TUI). The TUI is a browser based interface display that supports the presentation of any material that can be shown on a web-page. It can be displayed concurrently with a training application running, and has been shown to effectively house embodied pedagogical agents that delivered feedback information during a training event (Goldberg & Cannon-Bowers, 2015). However, a psychomotor domain of instruction is not the traditional use case. Setting up a laptop or display to house the TUI may not be feasible, so other solutions must be considered. In the case of EST training, this could mean modifying the EST software to display individualized visual feedback, per trainee, directly in front of them on the target screen. However, there could be 5-15 trainees receiving feedback simultaneously causing disruptive, perhaps even conflicting, visuals across the screen. Therefore it may be critical to ensure every trainee receives feedback in a way that does not distract the other trainees. This may include the application of tablet devices, where the TUI can be displayed to a trainee through a localized network. If the cost of tablet devices for all firing lanes is an issue, GIFT could potentially direct messages through an ear-piece or set of headphones that provides feedback through the auditory channel alone, which adheres to Moreno & Mayer’s (1999) modality principle. With the incorporation of personalization and communication modality factors in the instructional management decision process, there are many experiments to be run in the future that will test the effect of variations across the associated dimensions.

ADAPTIVE MARKSMANSHIP VALIDATION AND EXPERIMENTATION

Following the completion of all development tasks listed above, a full test-bed capability of GIFT managing a BRM training event will be available. This initial test-bed will incorporate a limited set of assessment capabilities as well as a limited set of instructional tactics that can be executed during runtime. The first assessment logic being integrated within GIFT for BRM is the expert descriptive models of behavior across the four fundamentals. These models will be integrated within GIFT to support real-time assessment, with each model defining what data is required and how to transform the data to support automated practices.

The next step is testing and validating the use of these models to effectively inform pedagogical decisions. We will run a large number of novice shooters sampled from a local university. As the novices perform BRM procedures, their performance and behavior will be compared against the expert representations to recognize any deviations from desired performance. Then we will be able to determine if error-specific feedback provided to the individual trainee will significantly improve their shot quality and consistency.

When deviations from expert behavior are recognized, GIFT will provide remedial content. The initial remedial content applied in this testbed will be video snippets of experts explaining and demonstrating the application of specific BRM fundamentals. This content will be configured for display on a mobile tablet that will be located next to the participating subject. Videos of this nature have been examined before in BRM related studies, with results showing their application to improve performance in an individualized training condition (Chung, Nagashima, Espinosa, Berka, & Baker, 2009). An initial experiment using our automated testbed will potentially involve the following conditions: (1) remediation informed by expert model assessment, (2) remediation selected randomly when performance does not meet performance criteria, (3) remediation informed by a live instructor, and (4) a control condition with no feedback in between groupings. This initial design will be used to assess the expert models’ utility at informing pedagogical interventions as well as the effect of personalized coaching on BRM performance.

In addition, will are also planning to collect expert annotation data of novice behavior across three separate individuals. This approach will support a feasibility evaluation for developing a buggy library that can provide further assessment capabilities. This data set will be analyzed post-hoc and will be treated as a supervised machine learning problem. We will then determine the strengths and weaknesses between the two domain model representations to identify any superiority that might influence the application of one over the other.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

In this paper, we presented current work involved with extending GIFT modeling and pedagogical techniques to support a psychomotor training domain; in this instance being BRM. This work is among the first to address the need for automated, personalized, and intelligent psychomotor training. Methods for measurement, model creation, expert validation, and novice diagnosis in this type of domain are addressed, along with their implications of being implemented in a domain-independent framework. This paper presents a way to measure items of interest, create models without a barrage of annotated or novice data, validate those models for reasonability, and apply them in a real context.

With an established workflow for building models and a testbed to run studies, future research will involve numerous components of marksmanship related training efforts. This includes investigating the application of these approaches on new weapons not associated with BRM, as well as investigating the feasibility of training more advanced skills (e.g., hitting moving targets). In addition, future research will also examine guidance within a psychomotor use case that will investigate variations in timing, specificity, and modality of feedback. Another area of research will focus on human performance related questions that involve elements of deliberate practice and sequencing of training tasks to better improve skill acquisition and retention.

REFERENCES

Amburn, C., Goldberg, B., & Brawner, K. (2014). *Steps Towards Adaptive Psychomotor Instruction.* Paper presented at the The Twenty-Seventh International Florida Artificial Intelligence Research Society (FLAIRS) Conference, Pensacola Beach, FL.

Chung, G. K., Nagashima, S. O., Espinosa, P. D., Berka, C., & Baker, E. L. (2009). An Exploratory Investigation of the Effect of Individualized Computer-Based Instruction on Rifle Marksmanship Performance and Skill. CRESST Report 754. *National Center for Research on Evaluation, Standards, and Student Testing (CRESST).*

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.

Ericsson, K. A. (2014). *The road to excellence: The acquisition of expert performance in the arts and sciences, sports, and games*: Psychology Press.

Goldberg, B., Amburn, C., Brawner, K., & Westphal, M. (2014). *Developing Models of Expert Performance for Support in an Adaptive Marksmanship Trainer.* Paper presented at the Interservice/Industry Training, Simulation, & Education Conference (I/ITSEC), Orlando, FL.

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*.

James, D. R., & Dyer, J. L. (2011). Rifle Marksmanship Diagnostic and Training Guide: DTIC Document.

Krathwohl, D. R. (2002). A revision of Bloom's taxonomy: An overview. *Theory into practice, 41*(4), 212-218.

Moreno, R., & Mayer, R. E. (1999). Cognitive principles of multimedia learning: The role of modality and contiguity. *Journal of Educational Psychology, 91*(2), 358.

Sottilare, R., Graesser, A., Hu, X., & Goldberg, B. (2014). Preface *Design Recommendations for Intelligent Tutoring Systems, Volume 2 - Instructional Management* (pp. i-xiv): U.S. Army Research Laboratory

Sottilare, R. A., Ragusa, C., Hoffman, M., & Goldberg, B. (2013). *Characterizing an Adaptive Tutoring Learning Effect Chain for Individual and Team Tutoring*. Paper presented at the The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC).

The U.S. Army Learning Concept for 2015. (2011). TRADOC Pamphlet 525-8-2

ABOUT THE AUTHORS

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).*

Mr. Charles Amburn *is the Senior Instructional Systems Specialist for the Advanced Simulation Branch of the Army Research Laboratory’s SFC Paul Ray Smith Simulation & Training Technology Center. He obtained a Film degree and a Master's degree in Instructional Systems Design from the University of Central Florida then began working in the Advanced Instructional Systems Branch at NavAir. There, he worked on special projects for the Navy and Marine Corps for 10 years before becoming the Lead Instructional Designer for the Army's Engagement Skills Trainer (EST) program at the Program Executive Office of Simulation, Training and Instrumentation (PEOSTRI), Orlando, FL.*

How to CREATE Domain Content for Intelligent Tutoring Systems.

Terry Patten[1], Max Metzger[1], Stephen Hookway[1], Amy Sliva[1], Samuel Lasser[1], Jeffrey Wallace[1], Rodney Long[2]
Charles River Analytics[1], U.S. Army Research Laboratory[2]

INTRODUCTION

A critical—but often overlooked—step in authoring intelligent tutoring systems is the initial retrieval and extraction of domain content. This step is increasingly challenging as the number and size of content resources grow rapidly. Enterprise repositories such as the Central Army Registry (CAR) and even the World Wide Web itself are valuable sources of domain content that should be easily exploited by authors of intelligent tutoring systems. Unfortunately, significant effort is currently required to transform this raw material into useful tutoring resources. In this paper we describe a framework for retrieving and extracting domain content that could serve as a useful addition to the Generalized Intelligent Framework for Tutoring (GIFT; Sottilare, Brawner, Goldberg, & Holden, 2012; Sottilare, Holden, Goldberg, & Brawner, 2013).

Content Retrieval and Extraction for Advanced Tutoring Environments (CREATE) is a prototype system that demonstrates several advanced technologies for turning raw materials into useful domain resources. CREATE uses the methods and standards of the Semantic Web (**http://www.w3.org/standards/semanticweb/**) to add rich semantic metadata that enables the material to be retrieved based on its *meaning* rather than on specific keywords. This semantic metadata also enables CREATE to recommend material based on the target learner's occupational specialty. CREATE automatically creates rich pedagogical metadata including readability scores, Component Display Theory quadrants (Merrill & Wood, 1974), Bloom levels (Krathwohl, 2002), and Gagne events (Driscoll, 2000). An important innovation in CREATE is that this rich metadata is attached to document *segments* so that tutor authors can find and retrieve useful chunks of material rather than whole documents, which are typically too large for authoring purposes. Each of the key CREATE technologies will be described in subsequent sections below, followed by a discussion of the CREATE architecture and concluding remarks.

RETRIEVING CONTENT WITH SEMANTIC SEARCH

Authors typically retrieve domain content with keyword searches. The limitations of keyword search are well understood and can be expressed in terms of the standard evaluation metrics for information retrieval systems: *precision* and *recall*. Precision is the percentage of returned material that is actually relevant to the query, and recall is the percentage of relevant material that is returned. Keyword search produces poor precision for two reasons: (1) keywords often have several meanings, so matches will occur in cases where the keyword is present in the document but has the wrong meaning; and (2) keywords from the query may all be present, but have no relationship to each other in the document. Keyword searches tend to produce poor recall because only documents that have the exact keywords will be matched—documents that contain synonyms or terms that are more specific than the keyword will be missed.

CREATE has been designed based on the principles and standards of the World Wide Web Consortium's (W^{3}C) Semantic Web project (Berners-Lee, et al., 200l; **http://www.w3.org/standards/semanticweb/**). The idea behind the Semantic Web is to search based on meanings rather than keywords. Searches based on meaning will return relevant material regardless of the terminology used in documents. The meanings or semantics are defined using an ontology—a hierarchy of concepts with relationships defined between the concepts. For example, the ontology may define the concept of a *survival radio* as equivalent to the concept of an *emergency transmitter*, and may define the AN/PRC-90 as a type of survival radio. A search for the concept *emergency transmitter* would match documents containing the concept *survival radio* (because the ontology has defined them to be equivalent) and documents containing the concept *AN/PRC-90* (because the AN/PRC-90 is a type of survival radio and that is equivalent to an emergency transmitter).

The W^{3}C standard language for defining ontologies is the Web Ontology Language, or OWL (**http://www.w3.org/OWL/**) and the standard for defining semantic relationships is the Resource Description Framework, or RDF (**http://www.w3.org/RDF**). CREATE is based on both of these standards. The search examples above illustrate how identifying semantic matches may require one or more inferences. An advantage of adopting the OWL and RDF standards is that open-source OWL reasoners, such as Pellet (Sirin, Parsia, Grau, Kalyanpur & Katz, 2007), are available to make these inferences.

Semantic search addresses the precision problem because the query will only match the material if the correct meaning is present—semantic searches distinguish between the different meanings a keyword may have. Semantic search addresses the recall problem because the OWL reasoner can infer equivalences and make class-subclass deductions, thereby finding material that would not be found with a keyword search. While semantic search offers significant benefits over keyword search, CREATE enables queries that contain any combination of semantic concepts and keywords (including queries that contain only keywords). This ensures that authors have a superset of the search capabilities that they currently use.

Automated Ontology Creation

One challenge for semantic content search is that it requires a domain ontology. Even for a limited domain, the ontology can be large, and existing ontologies are available for few domains in practice. Even if resources are available to design and build a domain ontology, that ontology must be updated over time as new concepts emerge and as terminology changes. For these reasons, CREATE employs several techniques to automate the building and maintenance of the domain ontologies that support semantic search.

The most straightforward technique for building an ontology is to import ontological knowledge from external resources. For example, a document or web page may list the parts of a particular piece of equipment in a table. Often, however, domain knowledge is embedded implicitly in unstructured text. For example, a sentence of the form, “The AN/PRC-90 and other survival radios...” indicates that AN/PRC-90 radios are a subclass of survival radios. This requires natural language processing software to parse the text and extract the class-subclass relationship. Unfortunately, language can be complex and this type of extraction will have a significant error rate. CREATE therefore verifies extraction results through crowdsourcing—asking users of the system to occasionally verify a particular semantic relationship. This is done through a banner at the top of a page that the user can ignore, which asks a yes/no question such as, “Is ‘AN/PRC-90’ a subclass of ‘survival radio’?” and provides the original sentence for context. These verifications can often be provided in one or two seconds. If several users verify the relationship, it can be inserted into the ontology. CREATE takes into account the characteristics of the user (e.g., Military Occupational Specialty, or MOS) and their query history when asking for verifications to ensure that these extractions are verified by people who are likely to know the correct answer. This crowdsourcing approach enables an ontology to be expanded and maintained quickly without placing a significant burden on any specific individuals.

Extracting Semantic Content Metadata

The metadata supporting semantic search can describe either entities or relationships. The semantics of entities enables searches for a particular piece of equipment (e.g., AN/PRC-90) to match other names for the same equipment including synonyms, nicknames, or NATO designators, for example. But semantic metadata can also describe specific relationships involving an entity, such as one between the entity *radio* and the action *repair*. This semantic relationship enables searching for documents that actually talk about radios being repaired as opposed to documents that happen to contain the keywords “radio” and “repair,” with no guarantee that the radio is what is being repaired. The actions of interest in an equipment domain may include *install*, *inspect*, *set up*, *prepare*, *check*, *operate*, *repair*, *monitor*, and *maintain*. The CREATE strategy is to use a natural language parser to look for sentences where one of these actions is the main verb and a piece of equipment is the object of that verb. This indicates that the action is being applied specifically to the equipment. The verbs and objects are contained in the domain ontology, so when a query asks for material on *radio maintenance*, a sentence that has *repair* as its verb and *AN/PRC-90* as its object will match the query.

Semantic Recommendations

Once the domain ontology is in place, there is still the challenge that users may not be familiar with its structure and therefore may not know what to ask for. To mitigate this challenge, CREATE contains an automated recommender system to help users navigate the semantic knowledge base by recommending semantic searches that might be of interest given the user’s current query and known search history. CREATE uses the same types of recommendation techniques that have been used successfully by commercial websites such as Amazon and Netflix. When the user queries for a particular concept, CREATE can suggest queries for related concepts in the ontology. For example, if a user searches for “Radio,” CREATE might recommend searching for specific categories of radios. This enables users to leverage the ontology without having to be intimately familiar with its details. CREATE also recommends alternative queries that may be interesting to the user given their current search and MOS (based on previous searches by users with the same MOS).

RETRIEVING CONTENT BY PEDAGOGICAL CATEGORIES

When searching for training materials, it is important to retrieve materials that have the right content, but it is also important to retrieve materials that are at the correct level of reading difficulty, the correct level of sophistication, and are appropriate for a particular stage of instruction. A learner looking for introductory material on a topic does not want examination questions or PhD-level discussions. Fortunately, pedagogical theories provide categories for determining whether material is appropriate for a given instructional context. CREATE assigns pedagogical metadata based on four example criteria: the Army's FORd, CAylor & STich (FORCAST) readability model; Merrill's Component Display Theory; Bloom's Taxonomy; and Gagne's Nine Events of Instruction. This metadata then enables users to search for materials that belong to a particular pedagogical category or combination of categories. In principle, CREATE can apply any pedagogical theory; these examples were chosen because they are widely used and can be viewed as complementary to each other (so applying them simultaneously is useful).

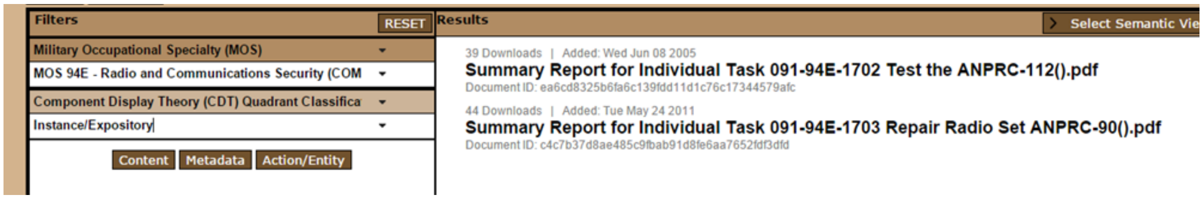
Extracting FORCAST Scores

This readability model was developed by the United States (U.S.) Army and estimates the school grade level for which the material is appropriate. Scores above 12 correspond to years of post-secondary education (e.g., 14 is sophomore college level). FORCAST scores are determined by the ratio of monosyllabic to polysyllabic words in the material and therefore easily computed automatically.

Extracting Component Display Theory Quadrants

Merrill's Component Display Theory is a pedagogical theory that (in its simplest form) assigns material to one of four categories (quadrants). An instructor either presents specific instances or presents generalities, and is either making statements or asking questions. Different combinations of these choices are appropriate in different learning situations. We performed an experiment where hundreds of examples of training materials from the Central Army Registry were assigned to one of the four quadrants by a former teacher. A support vector machine categorizer (Chang & Lin, 2011) was then trained and evaluated using a 10-fold cross-validation. A 10-fold cross-validation uses 90% of the data to train a model for each fold with the other 10% used for blind testing; this process is repeated 10 times and the accuracy scores on the blind data are averaged. Since all examples are used for blind testing, there is no possibility of the results being biased by which examples are chosen for testing.

The categorizer was able to assign quadrants to blind data accurately in three of the four cases: General/Expository (82%), Instance/Expository (88%), and Instance/Inquisitory (88%). General/Inquisitory had an accuracy of only 64% because there were too few examples of that category for training. Figure shows how searches for domain content can be filtered by Component Display Theory quadrants.



Military Occupational Specialty (MOS): MOS 94E - Radio and Communications Security (COM

Component Display Theory (CDT) Quadrant Classifica: Instance/Expository

39 Downloads | Added: Wed Jun 08 2005
Summary Report for Individual Task 091-94E-1702 Test the ANPRC-112().pdf
Document ID: ea6cd8325b6fa6c139fdd11d1c76c17344579afc

44 Downloads | Added: Tue May 24 2011
Summary Report for Individual Task 091-94E-1703 Repair Radio Set ANPRC-90().pdf
Document ID: c4c7b37d8ae485c9fbab91d8fe6aa7652fdf3dfd

Figure 1. Search including a Component Display Theory quadrant

Extracting Bloom Categories

Bloom's revised cognitive categories (Krathwohl, 1997) indicate the level of cognitive processing required by the material, ranging from remembering facts, to applying knowledge, to creating new knowledge. We obtained an academic data set where texts had been annotated with Bloom categories for a study (Zheng, Lawhorn, Lumley, & Freeman, 2008) of biology examinations such as the Medical College Admission Test (MCAT). The data included texts and the Bloom categories assigned by a group of experts in biological pedagogy. We trained a support vector machine categorizer using this data and performed a 10-fold cross validation. To our disappointment, the accuracy was only 52% on blind test cases (there were five categories present in the data, so the chance score is 20%). However, we then compared the individual assessments of the expert annotators against each other (which the data also provided) and discovered that any two of the human experts agreed with each other only 54% of the time on average (and annotators 1 and 3 only agreed 51% of the time). From this perspective our automatic categorization performed at the same level as the human experts.

Given that the human experts disagreed so often, it is clear that Bloom categorization is difficult. We suspect that the problem is that texts often contain elements of several different Bloom categories and that it may not be appropriate to limit the annotation to a single category. We therefore trained a separate binary categorizer for each of the three Bloom categories that had significant representation in the data set. If any of the three expert annotators placed a text in a specific category, the text was labeled as a member of that category. This means that many of the texts were assigned to multiple categories in both the training set and the test set. The accuracy for specific categories ranged from 57% to 75% in this case. This experiment suggests that the Bloom categorization task is quite difficult even if multiple categories can be assigned. Nevertheless, humans find Bloom assessments to be useful even if there is often disagreement about the details. Our human-level results suggest that it will be useful to assign Bloom categories to materials automatically.

Extracting Gagne Categories

Gagne categories (Driscoll, 2000) describe the nine steps that are typically required for successful instruction: starting with gaining the attention of the learner, through presenting the content, through assessing performance, and ending with enhancing retention. These categories are orthogonal to the Bloom categories: each of the Gagne categories can be applied to each of the Bloom categories (gaining the attention of novices versus gaining the attention of post-doctoral researchers; assessing the performance of novices versus assessing the performance of post-doctoral researchers); and each of these combinations provides very specific constraints on the material. We were not able to obtain a large experimental dataset for Gagne categories, but our initial experimental results indicate that the accuracy and issues will be very similar to those of Bloom categorization.

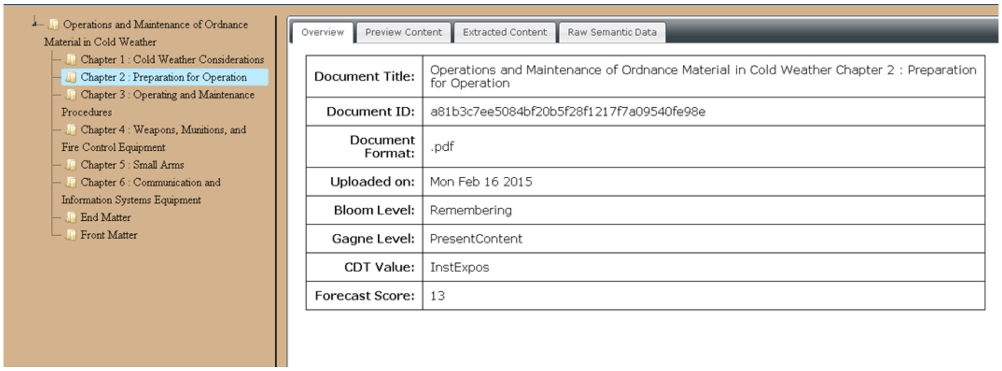
RETRIEVING CONTENT SEGMENTS

One of the innovative aspects of CREATE is that searches for domain content return only relevant segments of documents. The motivation for segmentation is that authors of tutoring materials do not want entire documents that may be many pages long—they want the paragraphs, sections, slides, and snippets that contain the relevant material. Segmenting documents is also important when assigning pedagogical metadata because different parts of a document may belong to different pedagogical categories (well-designed instructional material is likely to contain all of Gagne's categories, for example). Segmenting documents and applying pedagogical categories to those segments enables authors to retrieve just the introductory material or just the review questions for some domain concept.

The primary technical difficulty with segmenting documents is that each document type is formatted differently, so a set of segmentation rules must be formulated for each document type. However, documents of a particular type are formatted consistently, so once the appropriate segmentation rules are available, any number of documents of that type can be segmented quickly and easily.

In cases where the material is already broken up into explicit segments (chapters, sections, slides) the rules are straightforward; in cases where there are no clear text boundaries, segmentation can be quite difficult. Nevertheless, our research suggests that this type of document segmentation is feasible and can be automated completely in most cases. In difficult cases, a conservative segmentation strategy can be employed—in the worst case the user will end up with entire documents, which is exactly what current systems return.

Another difficulty when segmenting documents is providing the segments with meaningful names that can be displayed in the user interface. Documents typically have explicit titles, but document segments may not. In general, document segments should be given hierarchical names generated by concatenating labels. For example, if a document is entitled Modern Warfare, with a chapter entitled Ground War, the section can be given the label "Modern Warfare, Ground War." If strings are not available, sequential numbers can be used. Figure shows CREATE's summary of the categories extracted for a specific document segment; the segment's title was automatically created from the title of the parent document and the chapter title.



- Operations and Maintenance of Ordnance Material in Cold Weather
  - Chapter 1 : Cold Weather Considerations
  - Chapter 2 : Preparation for Operation
  - Chapter 3 : Operating and Maintenance Procedures
  - Chapter 4 : Weapons, Munitions, and Fire Control Equipment
  - Chapter 5 : Small Arms
  - Chapter 6 : Communication and Information Systems Equipment
  - End Matter
  - Front Matter

Document Title:	Operations and Maintenance of Ordnance Material in Cold Weather Chapter 2 : Preparation for Operation
Document ID:	a81b3c7ee5084bf20b5f28f1217f7a09540fe98e
Document Format:	.pdf
Uploaded on:	Mon Feb 16 2015
Bloom Level:	Remembering
Gagne Level:	PresentContent
CDT Value:	InstExpos
Forecast Score:	13

Figure 2. Summary of information extracted for a document segment

While there may be some difficult cases where significant effort is required to write the appropriate rules, our research indicates that documents can be segmented and those segments can be assigned reasonable names automatically.

THE CREATE ARCHITECTURE

Figure 3 shows the CREATE system architecture. Authors of intelligent tutoring systems interact with the CREATE system in two ways: through the CREATE Knowledge Exchange website or through third-party tools and applications (both shown on the far left of the diagram). Both these methods of interaction use Representational State Transfer (REST) web services to interact with CREATE. Third-party tools and applications can use the REST Application Programming Interface (API) to make SPARQL Protocol and Resource Description Framework (RDF) Query Language (SPARQL) queries directly to the SPARQL Query Engine, which is currently implemented using Apache Jena, with web services provided by Apache Fuseki (http://jena.apache.org). The Knowledge Exchange communicates through the CREATE REST Web Services to access additional CREATE features, including upload, extraction, and crowd-sourcing. The CREATE Web Services are implemented in Java Enterprise Edition, as well as Apache Tomcat (http://tomcat.apache.org).

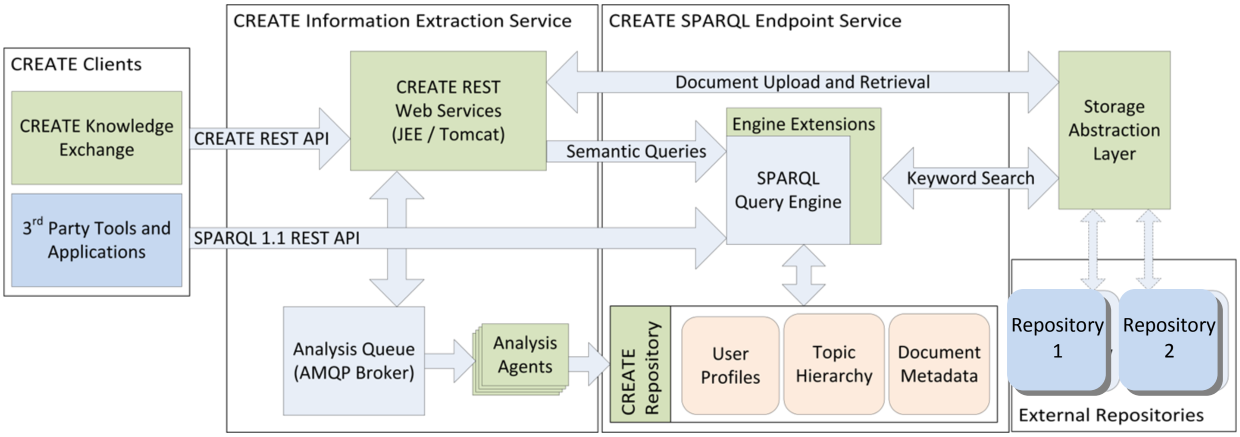

CREATE Clients
CREATE Knowledge Exchange
3rd Party Tools and Applications
CREATE REST API
SPARQL 1.1 REST API
CREATE Information Extraction Service
CREATE REST Web Services (JEE / Tomcat)
Analysis Queue (AMQP Broker)
Analysis Agents
CREATE SPARQL Endpoint Service
Document Upload and Retrieval
Semantic Queries
Engine Extensions
SPARQL Query Engine
Keyword Search
CREATE Repository
User Profiles
Topic Hierarchy
Document Metadata
Storage Abstraction Layer
Repository 1
Repository 2
External Repositories

Figure 3. CREATE Architecture

The following example demonstrates how the CREATE system architecture operates. A training manager wishes to add another U.S. Army training document to CREATE—the training manual TM 4-33.31 ("Operations and Maintenance of Ordnance Materiel in Cold Weather"). To do so, the training manager directs their web browser to the CREATE Knowledge Exchange website, where they log in via the authentication server (not shown) and then navigate to the upload page. The upload page then uses the CREATE REST Web Services to upload the document. The document itself is sent to the Storage Abstraction Layer where it is indexed for keyword-based search and stored within Apache Solr. Meanwhile, the document is also added to the Analysis Queue, where it is processed by various Analysis Agents. First, an agent identifies the document type (i.e., training manual) and sends it to the appropriate document segmenter. The document is split by chapter, with the preface, glossary, references, etc., also being separated, although smaller segments may be more appropriate for some types of documents. Each segment is stored in Apache Solr as well, so it can be retrieved by itself without retrieving the entire document. Each segment is then run through the suite of extractors and categorizers which extract a variety of metadata, such as actions performed on equipment and class-subclass relationships, which may expand the domain ontology after crowdsourced verification. As this process proceeds, the Analysis Queue sends progress updates to the Knowledge Exchange website.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

The initial retrieval and extraction of domain content is a critical and neglected aspect of authoring intelligent tutoring systems. We have described Content Retrieval and Extraction for Advanced Tutoring Environments (CREATE), a prototype system that demonstrates several new technologies. CREATE uses Semantic Web methods and standards to enable semantic searches for domain content—semantic searches retrieve appropriate content even if the query terms and document terms do not match exactly. Construction of the domain ontologies required for semantic search is accelerated by automatic extraction of ontological relationships from domain content, which are then verified by users through CREATE's crowdsourcing mechanism. Extraction of specific semantic relationships (for example between verbs and objects) produces metadata that enables precise semantic searches. CREATE automatically assigns pedagogical metadata using automatic categorizers that learn to recognize categories from training examples. Evidence from our experiments indicates that this pedagogical categorization is as accurate as human assessments. CREATE attaches content and pedagogical metadata to document segments, which are likely to be more useful to authors than whole documents.

CREATE demonstrates that the process of acquiring domain content can be automated to a large degree. Using these techniques, authors of intelligent tutoring systems can quickly acquire the domain content they need with the pedagogical properties they need, either in the form of segments of existing instructional material, or in the form of raw content (e.g., from the Central Army Registry) from which instructional material can be fabricated.

The techniques demonstrated by CREATE have the potential to improve the domain content of intelligent tutoring systems by ensuring that the most appropriate domain material is found by the author. Further, the techniques demonstrated by CREATE do not require tight integration with the specific mechanisms of an intelligent tutoring system—they can operate as a service that mediates between the author of an intelligent tutoring system and any set of content repositories. It is therefore recommended that CREATE be used as a model for a domain content retrieval and extraction service for GIFT (Sottilare et al., 2012).

ACKNOWLEDGEMENTS

The research reported in this document was performed in connection with contract/instrument W911QX-13-C-0072 with the U.S. Army Research Laboratory. The views and conclusions contained in this document/presentation are those of the authors and should not be interpreted as presenting the official policies or position, either expressed or implied, of the U.S. Army Research Laboratory or the U.S. Government unless so designated by other authorized documents. Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American*, May.

Chang, C-C. and Lin, C-J. (2011). "LIBSVM: A library for support vector machines." ACM Transactions on Intelligent Systems and Technology 2(3).

Driscoll, M. (2000) Gagné's Theory of Instruction. Chapter 10 in *Psychology of Learning for Instruction*. Needham Heights MA: Allyn and Bacon.

Krathwohl, D. (2002) A Revision of Bloom's Taxonomy, an Overview. *Theory into Practice*, Volume 41(4), 212-218.

Merrill, D. and Wood, N. (1974) Instructional Strategies: A Preliminary Taxonomy. ERIC Document Reproduction Service (No. SE018-771).

Sirin, E., Parsia, B., Grau, B., Kalyanpur, A., & Katz, Y. (2007). Pellet: A Practical OWL-DL Reasoner. *Journal of Web Semantics*, 5(2), 51-53.

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

Sottilare, R., Holden, H., Goldberg, B., & Brawner, K. (2013). The Generalized Intelligent Framework for Tutoring (GIFT). In C. Best, G. Galanis, J. Kerry, & R. Sottilare (Eds.) *Fundamental Issues in Defence Simulation & Training*. Ashgate Publishing.

Zheng, A., Lawhorn, J., Lumley, T., and Freeman, S. (2008). Assessment: Application of Bloom's Taxonomy Debunks the "MCAT Myth." *Science* 319, 414-415.

ABOUT THE AUTHORS

Dr. Terry Patten *is a Principal Scientist at Charles River Analytics. He leads research in semantic technology and natural language processing.*

Mr. Max Metzger *is a Software Engineer at Charles River Analytics. He specializes in tutoring systems and other training applications.*

Mr. Stephen Hookway *is a Software Engineer at Charles River Analytics. He specializes in applying semantic technologies and knowledge management techniques.*

Dr. Amy Sliva *is a Scientist at Charles River Analytics. She specializes in advanced semantic technology, behavioral models, and large-scale data analytics for decision making.*

Mr. Samuel Lasser *is a Scientist at Charles River Analytics. He specializes in computational linguistics including parsing and text categorization.*

Mr. Jeffrey Wallace *was an intern at Charles River Analtyics working on the CREATE Small Business Innovative Research (SBIR) project. He specializes in software development and integration.*

Mr. Rodney Long *is a Science and Technology Manager at the U.S. Army Research Laboratory-Human Research and Engineering Directorate Simulation and Training Technology Center (STTC). His interests include military training and intelligent tutoring systems.*

THEME V: GIFT USER PERSPECTIVES

The GIFT 2015 Report Card and the State of the Project

Keith Brawner[1], Scott Ososky[1, 2]
U.S. Army Research Laboratory[1], Oak Ridge Associated Universities (ORAU) [2]

INTRODUCTION

At the first GIFT Symposium, Benjamin Nye presented a paper titled "Toward a Generalized Framework for Intelligent Teaching and Learning Systems", which indicated that the Generalized Intelligent Framework for Tutoring (GIFT) was "heavier than required" and needed a more lightweight design (Nye & Morrison, 2013). In support of this vision, there were recommendations to break the explicit message passing format in the adaptive tutoring effect chain, simplify installation, and to provide "stub" examples of functionality. The GIFT development team has taken these recommendations seriously, and has addressed them directly through the simplification of installation and provision of example content across each of its releases.

While an amount of work has been performed to address those recommendations, solving some problems has created new ones. GIFT is multi-process and modular by its nature, which has resulted in a fragmented set of tools used to create tutors, where a single author may encounter many distinct authoring paradigms over the course of creation. Additionally, from a technical perspective, while GIFT no longer requires administrator privileges to install and operate, it still requires users to download an archive file (i.e., zip file) and install the software.

Further, the current release of GIFT is evolving toward a single, unified interface to author a tutoring system, completely controlled within a web browser. In a future version, to be released experimentally afterwards, an adaptive tutor created in this manner can additionally be provided to a student who needs only a browser to make use of the content. Likewise, information related to this student will be easily accessible to course facilitators. In this manner, GIFT will dramatically simplify the interface to author and consume tutoring, with plans to migrate GIFT "to the cloud", further simplifying the current authoring and delivery processes.

This paper has three distinct goals: 1) to make the community aware of the impact of their feedback on the development of GIFT, 2) reflect on the previous years of effort, and 3) to inform the community about the future directions of GIFT, including significant improvements to the authoring experience.

COMMUNITY-REQUESTED FEATURES

Firstly, for the members who are new to GIFT, the authors highly recommend reading the GIFT description paper (Sottilare, Brawner, Goldberg, & Holden, 2012). At the time of this writing, it still remains a relevant document describing the overall goals of the research program, system, and modules. A revised and updated document is in preparation, but the referenced work remains an accurate description for the high-level functionalities provided by each module and the eventual goals of utilization among learning locations. Additional high-level developmental information may be found in the

"Unwrapping GIFT" series of papers (Hoffman & Ragusa, 2014; Ragusa, Hoffman, & Leonard, 2013). Developers and super users are encouraged to read the documentation provided with each release of the GIFT (located at GIFT\docs\index.html). Finally, the GIFT forums, located at **https://gifttutoring.org/projects/gift/boards**, are actively monitored by a small team of developers, in addition to a series of Government project managers. One of the things colloquially noted in many conversations is that the GIFT forums "actually work". The forums, at the time of writing, have over 500 postings and responses.

In combination with suggestions gathered from the forums, there have been many suggested improvements in the previous meetings of the GIFT community (Sottilare, 2014; Sottilare & Holden, 2013). The recommendations have been documented in the associated proceedings, while being actively addressed by the development groups.

Table 1 (following page) summarizes the features and functionality requested during the previous years' GIFT Symposiums as completely as practical, with a summary of the degree to which each request has been or will be addressed. The following sections describe how each of the requests have been or are being addressed.

Separable Domain Information

The separation of domain logic from tutoring logic is one of the foundational principles of GIFT. One of the driving forces behind the development of GIFT is the need for many different subjects of training (medical, leadership, navigation, etc.), domains taxonomies (cognitive, affective, psychomotor, etc.), in many different environments (embedded, desktop, mobile, etc.), with diverse requirements (networked, online, offline, bandwidth-limited, etc.). Further requests have been made to personalize learning, which are being addressed through the dynamic generation and serving of HTML pages. GIFT has strived to separate domain-specific information from tutoring-specific information, and to better communicate domain-transferrable information, in support of these goals.

Error-Sensitive Feedback, Natural Language Feedback, Within-Game feedback

One of the first features requested for GIFT was natural language and within-environment feedback (Goldberg & Cannon-Bowers, 2013). The development team has met this request directly; the result is available to the community, and demonstrated in a variety of GIFT courses, such as the COIN and VBS2 instances. The current version of GIFT supports natural language feedback in a few manners: in the integration of AutoTutor services, scripts, in the authoring environment available in the "AutoTutor Script Authoring Tool" section of the Control Panel, and within any course with either an AutoTutor "mid-lesson survey", or AutoTutor script course element. The digital characters in the TC3Sim example courses demonstrate error-sensitive feedback both within and outside of the game environment.

Table 1 – A mapping of requested functionality, source of requests, and status of responses

	Functionality	Reference	Met	Plan
2.1	Separate domain-specific information from tutoring-specific information, better communicate domain-transferrable information	(Fancsali, Ritter, Stamper, & Berman, 2014; MacLellan, Wiese, Matsuda, & Koedinger, 2014)	X	X
2.2	Error-Sensitive Feedback, Natural Language Feedback, within-game feedback	(Goldberg & Cannon-Bowers, 2013)	X	
2.3	Unity environment	(Ray & Gilbert, 2013)	½	X
	Expand use of SIMILE through use of data model and expandable Gateway plug-in	(Mall & Goldberg, 2014)	X	
	Make further use of GAMETE	(Engimann et al., 2014)	X	X
2.4	Within-environment behavior detection	(Fancsali, Ritter, Stamper, & Nixon, 2013)	X	X
	Inferences for engagement and affect	(DeFalco & Baker, 2013; Fancsali et al., 2014; Rowe, Lobene, & Sabourin, 2013)	X	
	Transition prediction of states	(DeFalco & Baker, 2013; Rowe et al., 2013)	X	X
	Develop detectors and interventions of/for affective states	(Baker, DeFalco, Ocumpaugh, & Paquette, 2014; Fancsali et al., 2014)	X	
2.5	Survey scoring logic, links between domain and survey information, pedagogy based on this combined learner state, and other survey system improvements	(Pokorny, Chertoff, Holt, & Goldberg, 2014; Rowe et al., 2013; Sinatra, 2013)	X	
	Changes to the UserIDs, SAS, ERT, describe GIFT action reasoning	(Fancsali et al., 2014; Sinatra, 2014)	X	
2.6	Use the xAPI, expand its use in instructional tactics, integrate it with other systems	(Poeppelman, Hruska, Long, & Amburn, 2014)	X	X
	Use learner profile to support macro-sequencing, support hinting/remediation on content	(Long et al., 2014)	X	
2.7	Better installation, small suites of "utilities, wrappers, and stubs", remote procedure calls	(Nye & Morrison, 2013)	X	
2.8	Web-based authoring functions	(Hoffman & Ragusa, 2014; Rus et al., 2013)	X	
	Web-based, service-based	(Nye & Morrison, 2013; Rus et al., 2013)		X
	Gather requirements for ontology of messages to be made available, lighter weight	(Nye & Morrison, 2013)	On-going	
2.9	Communicate additional information to the Learner and Pedagogical	(Fancsali et al., 2014)		
	XNAgent platform for pedagogical agents	(Olney, Hays, & Cade, 2013)	*	*
	Use SimStudent to generate expert models	(MacLellan et al., 2014)		
	Conduct effectiveness analysis on authoring techniques	(MacLellan et al., 2014)		X

Unity

The integration of the Unity game environment, as requested at the first GIFT meeting (Ray & Gilbert, 2013), has had qualified success. A current, but unpublished/unreleased effort, Florida State University and Physics Playground (referred to as "Newtonian Talk") makes use of the Unity environment. Another effort with SRI International made use of the Unity game environment in a manner not able to be publicly released due to licensing and classification issues. Additionally, the Student Information Models for Integrated Learning Environments (SIMILE) authoring tool is able to process Unity game messages through its expanded use (Mall & Goldberg, 2014), and the SIMILE-using Game-based Architecture for Mentor-Enhanced Training Environments (GAMETE) effort has Unity among the environments where it enables interoperability, and is to be released to the community in the near term (Engimann et al., 2014). While Unity is not explicitly integrated into GIFT or currently available in the download package, *per se*, there is broad community support for its use, and it has been used in a variety of training settings.

Behavior Detection for Learner Modeling and Affect

Within-environment behavior detection for the purpose of advanced learner modeling and affect detection, has been mentioned in several papers (DeFalco & Baker, 2013; Fancsali et al., 2013; Rowe et al., 2013). GIFT has a unique problem in that the creation of a *single* collection, message passing routine, and interpretation workflow is difficult to create for *all* environments and *all* states. Among the findings while trying to implement this functionality is that the detection, collection, and interpretation of advanced learner information (traits, states, preferences, etc.) from inside of a training environment is similar to these functions outside of it (sensors, annotations, etc.), as requested similarly among the same research papers (DeFalco & Baker, 2013; Rowe et al., 2013). As such, the current version of GIFT has implemented a developmental version of this collection and detection from within and outside the training environment through the use of RapidMiner models which require no explicit programming to create or implement, and has demonstrated it at the AIED Conference (Rowe, Mott, & Lester, 2015). Further work to embed these processes in the web-based authoring environment for ease of creation is yet to be performed, although the total sequence *is* linked to instructional interventions, as requested in (Baker et al., 2014; Fancsali et al., 2014). Additional further work is needed to predict state transitions, which is now enabled after the measurement and communication of this state information.

Survey Authoring System Improvements

Several GIFT Symposium papers have requested improvements to the Survey Authoring System (SAS) (Pokorny et al., 2014; Rowe et al., 2013; Sinatra, 2013). Since these requests, several critical functions have been developed including: the ability to grade surveys, to use surveys as pre-test and post-test measures, to use the post-test measures as a manner of assigning remedial content, and more extensive back-end processing of survey results for ease of export into standard scientific tools such as IBM's SPSS, as requested in other works (Sinatra, 2014). Several sample surveys are now available in the SAS and surveys may be imported and/or exported. The authors recommend looking at the Dynamic Environments Testbed course, and the EMAP Simple Branching Example as sample ways to implement these survey system improvements in new experiments and courses. Those courses additionally show the implementation of a separate, yet related, recommendation to describe the reasoning behind each of the actions taken by GIFT's instructional engine.

Increased Learner Tracking

GIFT currently makes basic use of the Experience API (xAPI) Framework (Regan, 2013) in order to report all scored information from within a training session. GIFT only reports this information to a Learning Record System (LRS) if the Learning Management System (LMS) flag is configured to turn it on. This information is used in the GIFT course recommendations that a user sees upon first login, through the traversal of the stored xAPI statements. The development team intends to expand this functionality through an increase in fine-grained learner activity tracking, to use this information to provide remedial and supplemental training, and to recommend live training events.

Overall Simplification

Programmatically, there is a call for the overall simplification of the GIFT system (Nye & Morrison, 2013). Firstly, the authors note that GIFT is intended to be able to support a wide variety of training tasks, ranging from desktop "page-turners" to psychomotor marksmanship training (Goldberg, Brawner, Amburn, & Westphal, 2014), and is expanding to intelligent team tutoring. Development support of interfaces for physical sensors, diverse training environments, and team training environments is non-trivial. However, the current version of GIFT boasts a simplistic installation of extracting a compressed file and running a minor script. "Stubs", or examples of the creation of content, are created for several training environments, with interface stubs also available. Furthermore, there is a document provided in the current release of GIFT which diagrams the courses according to GIFT functionalities, allowing developers to determine, at-a-glance, the most profitable avenue to investigate for new developments.

Web Services and Hosting

Finally, there have been several members of the community who have generally argued for GIFT to be web-hosted, with browser-based authoring and delivery (Hoffman & Ragusa, 2014; Nye & Morrison, 2013; Rus et al., 2013). This is a technical challenge, one that will require time to implement, test, and stabilize. In our initial work, we took steps to enable adaptive course content to be administered over the web, and created browser-based interfaces for some of our authoring tools. In addition, the remaining components of GIFT have been modified to incorporate a real hosted system (Windows Server or Linux), in order to be able to manage content (without a file system), manage and track users (without assuming all users have access to all content and can make any changes), and interface with executable desktop applications and sensors through web-based interfaces. The upcoming Cloud based GIFT release leverages the Nuxeo framework for interactions, and has joined the Nuxeo community through its own GitHub page (**https://github.com/GIFT-Tutoring**). The GIFT Cloud release version will publish in the traditional server style with downloadable (VMs), and makes use of Java Web Start for interfacing with traditional desktop-based training solutions (e.g., PowerPoint, simulators, etc.).

More Improvements on the Horizon

There are a number of requested and recommended features for GIFT in Table 1 that have not yet been addressed, but are currently in planning. GIFT is currently in development of an agent-based framework for conversational engines. Most of this work has been open-sourced as part of the JChatScript project. GIFT has an interface in development to conversational agents which conform to the Extensible Messaging and Presence Protocol (XMPP) and JChatScript protocols within a Gateway specification. This technology will be demonstrated at the upcoming AIED conference (Brawner, 2015). The combination of those technologies seeks to address the same requirements as originally stated in the paper requesting an XNAgent solution (Olney, Hays, & Cade, 2013). With respect to the latter requests in Table 1: additional collaboration and support is needed to leverage SimStudent (MacLellan et al., 2014) in GIFT, and to communicate additional information to the learner and pedagogical modules (Fancasli et al. 2014).

Finally, members of the GIFT community have concurrently identified the need to examine the effectiveness of current authoring techniques. Authoring, for the purposes of the current document, represents those activities associated with the design of adaptive course flow, collection and management of course materials, and configuration of external systems including sensors and training environments. Significant progress has been made in GIFT authoring, starting as a collection of editable eXtensible Markup Language (XML) files, and evolving toward more user-friendly graphical web-based interfaces. Graphical interfaces alone, however, do not wholly address the needs of an authoring solution. There also exists a need to examine the underlying processes associated with authoring an adaptive tutor, and how those processes drive the requirements for authoring tools. Authoring is an area that will continue to receive attention and effort, as GIFT transitions to cloud-based environments, supported by web-based interfaces. The following section further examines the state of authoring processes and tools, as well as our approach for meeting the needs of GIFT's user community in developing a useful authoring solution.

IMPROVEMENTS IN AUTHORING PROCESSES AND TOOLS

Adaptive Authoring Processes

Adaptive tutor authoring, as a process, is in its infancy. Adaptive tutors and linear, computer-based training may bear some superficial similarities in their *delivery* and *presentation*; however the inherent complexity of an adaptive tutor is many orders of magnitude greater due to the inclusion of learner models, pedagogical agents, sensor configuration and adaptive content management. Thus, adaptive tutor authoring, as a *content creation activity*, represents a relatively new interaction paradigm, one that is not yet clearly defined. Adaptive tutor authoring processes might be similar to other content creation activities, such as building a slide deck, designing a blog post, or writing a piece of interactive narrative. Though, none of these activities represent an exact one-to-one mapping to authoring adaptive content, given the unique elements described above. As such, we approach adaptive tutor authoring as a *new* form of content creation, with *processes* that need to be well-defined and potential *skills* that authors will need to develop.

Mental models serve as the underlying theory in our user-centered approach to defining adaptive tutor authoring processes. Rouse and Morris (1986) described mental models as “the mechanisms whereby humans are able to generate descriptions of purpose and form, explanations of system functioning and observed system states, and predictions of future states” (p. 7). Further, potential authors must form *new* mental models regarding the processes though which adaptive tutors are created. Those new mental models will be formed by integrating new knowledge (of adaptive tutoring) with existing mental models of those activities that are perceived to be similar (e.g., designing a web page, building a slide deck).With mental models, we can begin to identify what potential authors need to understand about adaptive tutoring systems, as well as the depth and accuracy of authors’ existing knowledge of those systems. It is also important to consider the various ways in which potential authors *expect* to build adaptive tutors, because mental models do not need to be complete or even accurate in order for users to apply them to system interaction (Norman, 1986).

One of the central tenants of GIFT is to reduce the skill and resources required to develop an intelligent tutor. By adhering to this tenant, we hope to make GIFTs capabilities available to the broadest audience possible. Potential authors may interact with our authoring system with a variety of different backgrounds, including instructional design, experimental research, and subject matter expert. Through literature review, interviews, and ethnographic observation, we intend to illustrate users’ mental models of adaptive tutor design, as well as non-system-specific goals of those potential users. Outputs from this effort will help to identify differences in the needs between each user group, which will aid in the identification and development of features that can be implemented into GIFT in order to provide greater support for GIFT’s authors. We will also leverage users’ goals, desires, and intentions to drive the requirements for the design of improved authoring tools, terminology, and interfaces for GIFT.

Authoring Tools and Usability

Understanding potential authors’ mental models will aid the development team in designing features and authoring interfaces. Looking toward the future, we might imagine GIFT authoring consisting of a number of different experiences for novices, general users, and power users, respectively. However, it is unlikely, at least in the near term, that it will be possible to design-out all of the complexity of authoring an adaptive tutor. Authors will likely need to apply new and/or existing skills and knowledge in order to use the authoring tools. Additionally, the authoring tools should not be so simplistic as to limit the potential of GIFT, but not so complex that authors abandon the task in frustration. Thus, we must strive for reasonable balance in the tradeoffs between *usability, depth,* and *flexibility*.

Until recently, GIFT applications were created by writing and/or editing extensible markup language (XML). The current stable and experimental versions of GIFT (2014-2 and 2014-3X, respectively) provide browser-based interfaces to facilitate semi-automated user creation of the XML output. However, a heuristic evaluation of the current set of GIFT authoring tools revealed that the experience across each of these distinct interfaces is inconsistent, and those interfaces use technical and system level language, some of which is specific to GIFT. Thus, not only do potential authors need to have a deep understanding of adaptive tutoring, the current design of the interface also requires users to have a detailed understanding of the inner workings of components within GIFT. This may result in situations in which authors may know what they want to build, but are unable to translate their intentions into actions with the current set of GIFT authoring tools. For the next planned release of GIFT (2015-1), development effort is being directed toward creating a unified, consistent interface across all of GIFT's authoring tools, as well as using human-understandable labels and language that promote system learnability with authors. These represent important steps toward making GIFT more accessible to a broader audience.

In support of authors, we also intend to conduct formative evaluations of GIFT, concurrent with the engineering development of authoring tools, processes, and interfaces. Our goal is to continuously and rapidly collect real user feedback in order to more quickly implement authoring improvements and establish a working dialogue with the GIFT authoring community. Our planned approach consists of three major thrusts: usability evaluation, site analytics, and user feedback. Usability evaluations are conducted with members of the GIFT community in a variety of critical authoring tasks in order to identify failure points within GIFT's authoring tools, as well as the reasons why the failure occurred. With a better understanding of authors' needs and expectations, we can structure the authoring experience from a user-centered perspective. We also intend to implement a site analytics package on GIFT's web-based authoring tools in an effort to better understand quantitative aspects of GIFT's authoring tools, including the time required for authoring a tutor, modeling an author's navigation through the tools, and if/where authors abandon the authoring task. We can leverage quantitative data in order to make improvements that reduce the time to author a tutor, structure the layout of the tools in an intuitive manner, and provide on-demand help and documentation at the point of need. Finally, we plan to embed a feedback mechanism into the GIFT authoring tool suite. The feedback function will provide another way to communicate with the GIFT development team, in addition to the issue (bug) tracking system and traditional support channels. The purpose of the feedback function is to provide an easy way (i.e., on demand, in application) for the GIFT authoring community, at large, to send a quick note to the development team for further consideration. These quick feedback notes might consist of ideas for new features, sections of the tools that are confusing, and/or a subjective rating of a specific tool or page. Overall, we intend to generate a holistic assessment of GIFT's authoring tools, in order to guide the future development of authoring tools as GIFT moves beyond the desktop into the cloud environment.

Authoring in the Cloud

GIFT, in total, will be moving to a cloud-based environment. This feature will be implemented throughout the next few releases of GIFT. Before describing the potential impact of this transition on the authoring experience, it is important to point out that the existing ability to download and install GIFT on a computer will still be available for the foreseeable future. The GIFT development team recognizes that there may exist a need or preference for some user groups to be able to operate GIFT on a private network, or in an offline capacity, for a number of different reasons. Or, perhaps an author prefers to use GIFT's online capabilities, but needs to transfer their work to an offline environment for a time due to travel, internet availability and so on. As part of our efforts to continuously improve the GIFT authoring experience, we are working to maintain an awareness of these use cases, and provide support for such users to the best of our abilities.

The transition of GIFT to the cloud will see a transition in the authoring tools from a desktop application to a browser-based tool. This creates a significant opportunity for a number of important improvements to the authoring experience. One immediate benefit of a browser-based authoring environment is the elimination of the need to download software. For instance, GIFT's current software installation experience, while much improved, still requires a number of steps to locate the software online, unpack the archive, and run the install. A cloud-based environment also enables the possibility of online storage of course structure and supporting content, which would allow authors to work across systems and collaborate with other members of an *authoring team*. Along with online storage, a cloud-based GIFT also provides for the ability to integrate with other relevant systems, including content management and long-term learner record storage. Finally, the nature of a persistent, online environment enables the development team to issue feature updates and bug fixes to the authoring community in a manner that is less-intrusive, reduces the end-user burden to update software, and allows for more timely updates in response to feedback.

CONCLUSION

At the first GIFT Symposium, it was reported that GIFT was an avenue for researchers to transition research from an academically-based setting to a use-based setting, and that there exists several options to do so (Brawner, 2013). Although the GIFT project is growing, it remains a highly collaborative effort, which responds to the needs of its community. The community, in the past, has suggested a large number of features for inclusion in the developmental baseline; the vast majority of which have been developed and provided back. The features which have not been included are either planned or subsumed by competing equivalent technologies. The authors, when creating Table 1, were pleased that the needs and requests of the community were being met.

The path forward

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. There are a number of significant features on the horizon for future versions of GIFT, which include training that extends to teams (and teams of teams), collaborative authoring, additional pedagogical engines, intelligent agents, and multi-platform support. As such, GIFT will be continuously improved and developed for the foreseeable future. The members of the GIFT community 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 useful tools, powerful technologies, and positive user experiences for next year's releases and beyond.

REFERENCES

Baker, R. S., DeFalco, J. A., Ocumpaugh, J., & Paquette, L. (2014). *Toward Detection of Engagement and Affect in a Simulation-based Combat Medic Training Environment* Paper presented at the GIFTSym2, Pittsburgh, PA.

Brawner, K. (2013). *GIFT Research Transition: An Outline of Options*. Paper presented at the GIFT Symposium 2013, Memphis, TN.

Brawner, K. (2015): *Rapid Dialogue and Branching Tutors*. Artificial Intelligence in Education 2015, Madrid, Spain

DeFalco, J. A., & Baker, R. S. (2013). *Detection and Transition Analysis of Engagement and Affect in a Simulation-based Combat Medic Training Environment.* Paper presented at the AIED 2013 Workshops Proceedings Volume 7.

Engimann, J., Santarelli, T., Zachary, W., Hu, X., Cai, Z., Mall, H., & Goldberg, B. (2014). *Game-based Architecture for Mentor-Enhanced Training Environments (GAMETE)*. Paper presented at the GIFTSym2, Pittsburgh, PA.

Fancsali, S., Ritter, S., Stamper, J., & Berman, S. (2014). *Personalization, Non-Cognitive Factors, and Grain-Size for Measurement and Analysis in Intelligent Tutoring Systems: Implications for GIFT*. Paper presented at the GIFTSym2, Pittsburgh, PA.

Fancsali, S., Ritter, S., Stamper, J., & Nixon, T. (2013). *Toward "Hyper-Personalized" Cognitive Tutors.* Paper presented at the AIED 2013 Workshops Proceedings Volume 7.

Goldberg, B., Brawner, K., Amburn, C., & Westphal, M. (2014). *Developing Models of Expert Performance for Support in an Adaptive Marksmanship Trainer*. Paper presented at the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC), Orlando, FL.

Goldberg, B., & Cannon-Bowers, J. (2013). *Experimentation with the Generalized Intelligent Framework for Tutoring (GIFT): A Testbed Use Case.* Paper presented at the AIED 2013 Workshops Proceedings Volume 7.

Hoffman, M., & Ragusa, C. (2014). *Unwrapping GIFT: A Primer on Authoring Tools for the Generalized Intelligent Framework for Tutoring*. Paper presented at the GIFTSym2, Pittsburgh, PA.

Long, R., Amburn, C., Barnieu, J., Hyland, S. J., Zoellick, C. E., & Beauchat, T. A. (2014). *Adaptive Training in the Soldier-Centered Army Learning Environment (SCALE)*. Paper presented at the GIFTSym2, Pittsburgh, PA.

MacLellan, C. J., Wiese, E. S., Matsuda, N., & Koedinger, K. R. (2014). *SimStudent: Authoring Expert Models by Tutoring*. Paper presented at the GIFTSym2, Pittsburgh, PA.

Mall, H., & Goldberg, B. (2014). *SIMILE: An Authoring and Reasoning System for GIFT*. Paper presented at the GIFTSym2, Pittsburgh, PA.

Norman, D. A. (1983). Some observations on mental models. In D. Gentner & A. L. Stevens (Eds.), *Mental Models*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Nye, B. D., & Morrison, D. (2013). *Toward a Generalized Framework for Intelligent Teaching and Learning Systems.* Paper presented at the Artificial Intelligence in Education (AIED).

Olney, A., Hays, P., & Cade, W. (2013). *XNAgent: Authoring Embodied Conversational Agents for Tutor-User Interfaces.* Paper presented at the Artificial Intelligence in Education (AIED).

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.

Pokorny, B., Chertoff, D., Holt, L., & Goldberg, B. (2014). *Structure of Instructional Interactions Within GIFT.* Paper presented at the GIFTSym2, Pittsburgh, PA.

Ragusa, C., Hoffman, M., & Leonard, J. (2013). *Unwrapping GIFT: A Primer on Developing with the Generalized Intelligent Framework for Tutoring*. Paper presented at the GIFT Symposium, Memphis, TN.

Ray, C., & Gilbert, S. (2013). *Bringing Authoring Tools for Intelligent Tutoring Systems and Serious Games Closer Together: Integrating GIFT with the Unity Game Engine.* Paper presented at the AIED 2013 Workshops Proceedings Volume 7.

Regan, D. A. (2013). *The Training and Learning Architecture: Infrastructure for the Future of Learning.* Paper presented at the Invited Keynote International Symposium on Information Technology and Communication in Education (SINTICE), Madrid, Spain.

Rouse, W. B., & Morris, N. M. (1986). On looking into the black box: Prospects and limits in the search for mental models. *Psychological Bulletin, 100*(3), 349.

Rowe, J., Lobene, E. V., & Sabourin, J. (2013). *Run-Time Affect Modeling in a Serious Game with the Generalized Intelligent Framework for Tutoring.* Paper presented at the AIED 2013 Workshops Proceedings Volume 7.

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.

Rus, V., Niraula, N., Lintean, M., Banjade, R., Stefanescu, D., & Baggett, W. (2013). *Recommendations For The Generalized Intelligent Framework for Tutoring Based On The Development Of The DeepTutor Tutoring Service.* Paper presented at the Artificial Intelligence in Education (AIED).

Sinatra, A. (2013). *Using GIFT to Support an Empirical Study on the Impact of the Self-Reference Effect on Learning.* Paper presented at the AIED 2013 Workshops Proceedings Volume 7.

Sinatra, A. (2014). *The Research Psychologist's Guide to GIFT*. Paper presented at the GIFTSym2, Pittsburgh, PA.

Sottilare, R. A. (2014). *GIFTSym2 Proceedings.* Paper presented at the GIFTSym2, Pittsburgh, PA.

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

Sottilare, R. A., & Holden, H. K. (2013). *Recommendations for Authoring, Instructional Strategies and Analysis for Intelligent Tutoring Systems (ITS): Towards the Development of a Generalized Intelligent Framework for Tutoring (GIFT).* Paper presented at the Artificial Intelligence in Education 2013, Memphis, TN.

ABOUT THE AUTHORS

***Keith Brawner, PhD** is a researcher for the Learning in Intelligent Tutoring Environments (LITE) Lab within the U. S. Army Research Laboratory's Human Research & Engineering Directorate (ARL-HRED, and is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT). He has 9 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.*

***Scott Ososky, PhD** is a Postdoctoral Research Fellow at the Simulation & Training Technology Center (STTC) within the U.S. 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 tools and interfaces within the adaptive tutor authoring workflow. 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.*

The Instructor's Guide to GIFT: Recommendations for using GIFT In and Out of the Classroom

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

INTRODUCTION

The Generalized Intelligent Framework for Tutoring (GIFT) has many useful functions for instructors and teachers who wish to use it in their classes. GIFT, which can be downloaded for free from www.gifttutoring.org, can be beneficial in both in-person lecture courses (serving as homework or external assignments) and in online courses which focus on self-paced learning. Today there are many college courses that are web-based, or include a web-component. This creates issues that are different than in traditional classrooms. Some of the primary challenges for students are the regulation and monitoring of their own learning, and engagement with the material (Lim, 2004). Intelligent tutoring systems (ITSs) are an ideal way to provide interactive material that adapts to the individual student, and can help in engaging the student with the material. However, most ITSs are closely tied to the topic that they were created with, and do not have tools that are flexible enough for a beginner to use. GIFT is a domain-independent framework for creating ITSs; It has a suite of tools that are usable in their current state, and are being improved to be more user-friendly (Sottilare, Brawner, Goldberg, & Holden, 2012). GIFT has been demonstrated to be useful for research psychologists, and in the development of experiments (Goldberg & Cannon-Bowers, 2013; Sinatra, 2013; Sinatra, 2014; Sinatra, Sims, Sottilare, 2014). Similarly, GIFT is a powerful tool for instructors who wish to create full ITSs for their students to use, or create course materials. Additionally, GIFT has many functions and features that can be adapted for use by instructors even if the adaptive tutoring capabilities are not fully used. The current paper provides recommendations on how GIFT can be used in in-person and online classes. Additionally, it discusses some of the practical concerns of using GIFT in this manner, and provides recommendations on future features that would improve GIFT's functionality for these tasks.

USING GIFT IN THE TRADITIONAL CLASSROOM

There are a number of different ways that instructors can utilize GIFT in the traditional classroom, where the main focus of the course is a lecture. The most traditional function would be to create course material and use GIFT to provide it to students. This can be done by having the instructor gather their course material, use GIFT's authoring tools to create a GIFT course, and then export the course. The exported version of the GIFT course can then be provided to the students so that they can download and install it on their own individual computer. Many instructors use PowerPoint presentations in their classes, and have a very good understanding of how to create materials using this medium. As GIFT supports PowerPoint shows, instructors can use their premade PowerPoints with GIFT or create new ones. Additionally, if the instructor wishes to include quizzes with their course for either assessment or learning purposes he or she can author questions and create "surveys" using GIFT's Survey Authoring System (SAS).

Advantages to Using GIFT in the Traditional Classroom

Instructor created GIFT courses can be used for many purposes. Firstly, they can simply be used as a way to present information to the students. Secondly, they can be used to provide review material and quizzes that can assist the students in studying for upcoming exams. Thirdly, they can be used for homework assignments that review and build on the in-class learning material. Fourthly, they could be a means of providing course activities that are engaged in during a dedicated computer lab time.

The GIFT authoring tool provides a course author with the ability to put together a linear course that has a number of different "transitions" or learning components. The course links together all of the materials, quizzes, and web-sites that the author has provided. An example course flow would be a welcome message and introduction to the course, a multiple choice pre-test, a PowerPoint presentation, a multiple choice quiz reviewing the newly learned concepts, another PowerPoint presentation, and a final post-test quiz. Simple courses can be authored without needing knowledge of computer programming, or advanced experience with GIFT. GIFT can also provide links to external web-sites, and PDFs as part of the course flow. Therefore, students can be presented with material of a number of different file types in a sequential order. Further, more advanced users may want to create a DKF (Domain Knowledge File) in order to allow for adaptive feedback to individual students based on performance. At current time a DKF needs to be associated with a training application (such as PowerPoint). In the case of PowerPoint, the time spent on each slide can be monitored and auditory feedback indicating that the students should slow down or speed up can be provided. However, adaptive training associated with survey performance can also be authored using the EMAP (Engine for Management of Adaptive Pedagogy) which allows for specific concepts to be assessed via authored question banks, which will trigger remediation that is specific to the individual's performance. It is important to note that the remediation provided by the EMAP is based on established pedagogy and is automatic rather than author directed. However, the author will tag the appropriate materials that should be provided for remediation with relevant metadata regarding their difficulty and mode that is used by the EMAP to provide content.

Regardless of the method that is used to integrate GIFT into the classroom, it will provide individualized material to students that he or she can directly interact with. This also allows the student to go at his or her own pace with the material and to review as they want. This is an innovative and engaging way to create a homework assignment or to provide self-study opportunities to students.

Challenges to Using GIFT in the Traditional Classroom

In its current form, using GIFT in this manner has a few challenges. Among these challenges are the need for individual students to download and install GIFT, the need for students to import courses into GIFT, and for the results of GIFT assessments to be communicated to the instructor.

At the current time there is no perfect strategy to provide the output of the course to the instructor so that they can assess performance or provide grades based on it. GIFT outputs a log file that can be copied from the computer that the course was taken on, then provided to the instructor to be extracted in GIFT's Event Reporting Tool (ERT). However, this would require the student to find the correct log and send it to the instructor. Further, the log file is unlocked, and is editable by the student. Therefore, a student could have the opportunity to change his or her answers in the log file before providing it to the instructor. This makes it difficult to trust the output of GIFT surveys for grades, as savvy students may try to change the answers they provided to the correct ones. For this reason it may be most advantageous to use questions in GIFT as points for completion or for review as opposed to a source of major course grades.

While courses that have been exported as tutors from GIFT are fairly straightforward to install, some students may run into challenges or have computers with operating systems that are not currently supported by GIFT (e.g., Apple). In the near future, GIFT will be moving to the Cloud. By having a Cloud option for students it will make things more straightforward, however, teacher and student roles will not necessarily be separated in the early Cloud versions. The different roles and permissions of these roles will likely develop over time, and would benefit from GIFT user input regarding the requirements and preferences for these roles. Once these roles are further developed, it will assist in the workflow for providing student logs to the instructors. It would be beneficial for the logs to be saved somewhere that the instructor can access or be automatically sent to the instructor without any student intervention or the possibility of them being edited.

USING GIFT IN THE ONLINE CLASSROOM

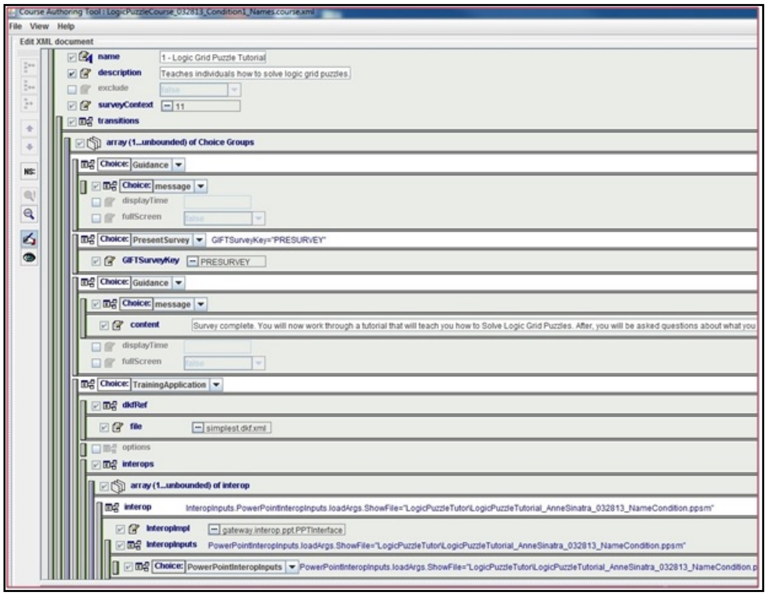
While GIFT is beneficial for teachers who offer traditional lecture based courses, it can additionally have use for online only and mixed mode classes. The majority of the suggested strategies and challenges that apply to using GIFT as part of a traditional lecture based class also apply to using GIFT in the online classroom. However, there are additional challenges present in online courses, as the student needs to be engaged and regulate his or her own learning. GIFT’s upcoming Cloud versions will be particularly helpful in online only courses, as students are used to interacting with online systems for course materials.

In an online only environment it is important to engage students actively in the learning process and help them to retain the information that they have learned. Therefore, developing downloadable or accessible GIFT courses that are interactive can help solidify the material that the student is reading about on his or her own. While adaptive feedback is not able to be provided in GIFT's current state by using only a DKF and the SAS, adaptive and interactive tutorials can be created using PowerPoint and Visual Basic for Applications (VBA). An example of such a course is available with GIFT (Logic Puzzle Tutorial). Additional resource such as the books *Powerful PowerPoint for Educators: Using Visual Basic for Applications to Make PowerPoint Interactive* (Marcovitz, 2012) and *PowerPoint for Teachers: Dynamic Presentations and Interactive Classroom Projects* (Finkelstein & Samsonov, 2007) offer suggestions and activities that can be used as a starting point for instructors who are new to programming or have a little bit of programming experience. These interactive PowerPoint files use macros and the output of the files can be written to excel files if assessment and saving of the output is desired. However, this has the same challenge as GIFT log files as they can be changed by students before being passed onto the instructor. Further, if students are required to engage with courses that include these files it is necessary for them to have both Excel and PowerPoint, which some students may not have on their personal computers.

TOOLS FOR INSTRUCTORS USING GIFT

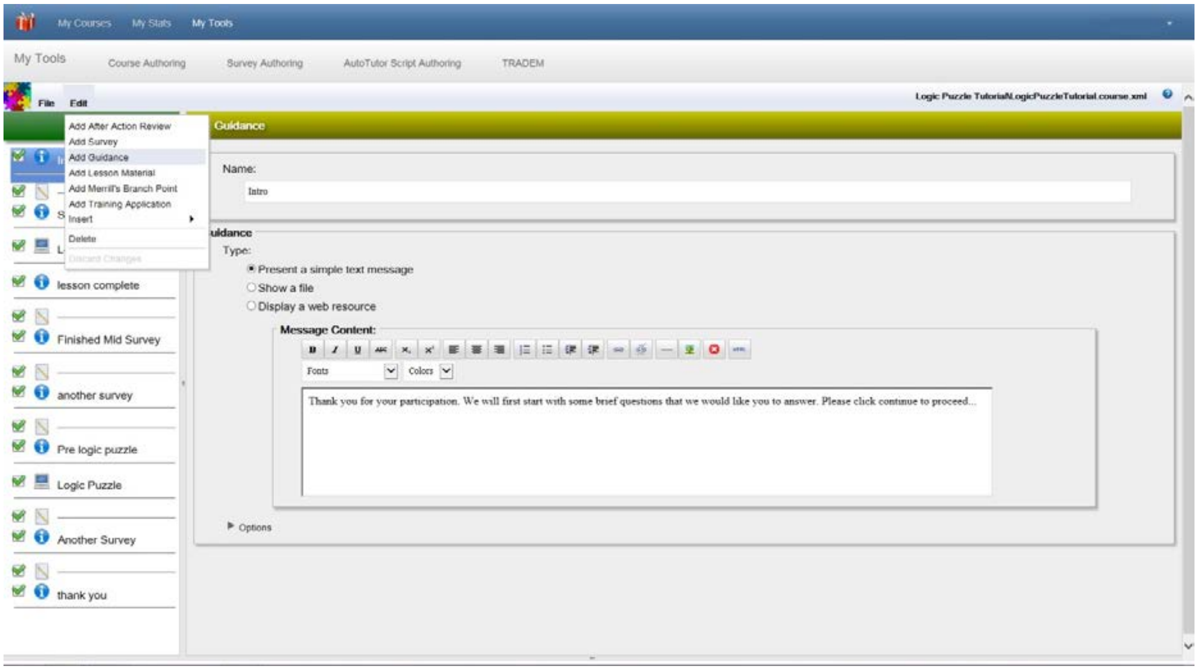
Course Authoring

GIFT has a number of different authoring tools that are of use to instructors who are creating GIFT content for their classes. Among them are the Course Authoring Tool (CAT) and GIFT Authoring Tool (GAT). The CAT is the original authoring tool that was included in GIFT to create courses. It is in the form of an .xml file editor and provides a linear representation of the course that is being created. On first glance it is not familiar to many users, but if they read about it and work with it, the design is easy to understand. The CAT is still included in GIFT and is a great resource for course instructors. The GAT is also included with GIFT and while it includes the same functionality as the CAT, it provides a more user-friendly authoring interface. See Figure 1 for a course loaded in the CAT. GIFT and the GAT have been undergoing authoring tool redesign, and moving towards a GIFT Dashboard interface. Figure 2 demonstrates the same course as Figure 1 loaded in the updated GAT which will be available with GIFT-2015-1.


Course Authoring Tool : LogicPuzzleCourse_032813_Condition1_Names.course.xml
File
View
Help
Edit XML document
NS:

☑ name 1 - Logic Grid Puzzle Tutorial
☑ description Teaches individuals how to solve logic grid puzzles
☐ exclude false
☑ surveyContext 11
☑ transitions
☑ array (1...unbounded) of Choice Groups
Choice: Guidance
☑ Choice: message
☐ displayTime
☐ fullScreen false
Choice: PresentSurvey GIFTSurveyKey="PRESURVEY"
☑ GIFTSurveyKey PRESURVEY
Choice: Guidance
☑ Choice: message
☑ content Survey complete. You will now work through a tutorial that will teach you how to Solve Logic Grid Puzzles. After, you will be asked questions about what you
☐ displayTime
☐ fullScreen false
Choice: TrainingApplication
☑ dkfRef
☑ file simplest.dkf.xml
☐ options
☑ interops
☑ array (1...unbounded) of interop
interop InteropInputs.PowerPointInteropInputs.loadArgs.ShowFile="LogicPuzzleTutor\LogicPuzzleTutorial_AnneSinatra_032813_NameCondition.ppsm"
☑ interopImpl gateway.interop.ppt.PPTInterface
☑ interopInputs PowerPointInteropInputs.loadArgs.ShowFile="LogicPuzzleTutor\LogicPuzzleTutorial_AnneSinatra_032813_NameCondition.ppsm"
☑ Choice: PowerPointInteropInputs PowerPointInteropInputs.loadArgs.ShowFile="LogicPuzzleTutor\LogicPuzzleTutorial_AnneSinatra_032813_NameCondition.p

Figure 1. A course loaded in the CAT, .xml file editor authoring tool. Note the linear structure of the course.



Name:

Intro

uidance

Type:

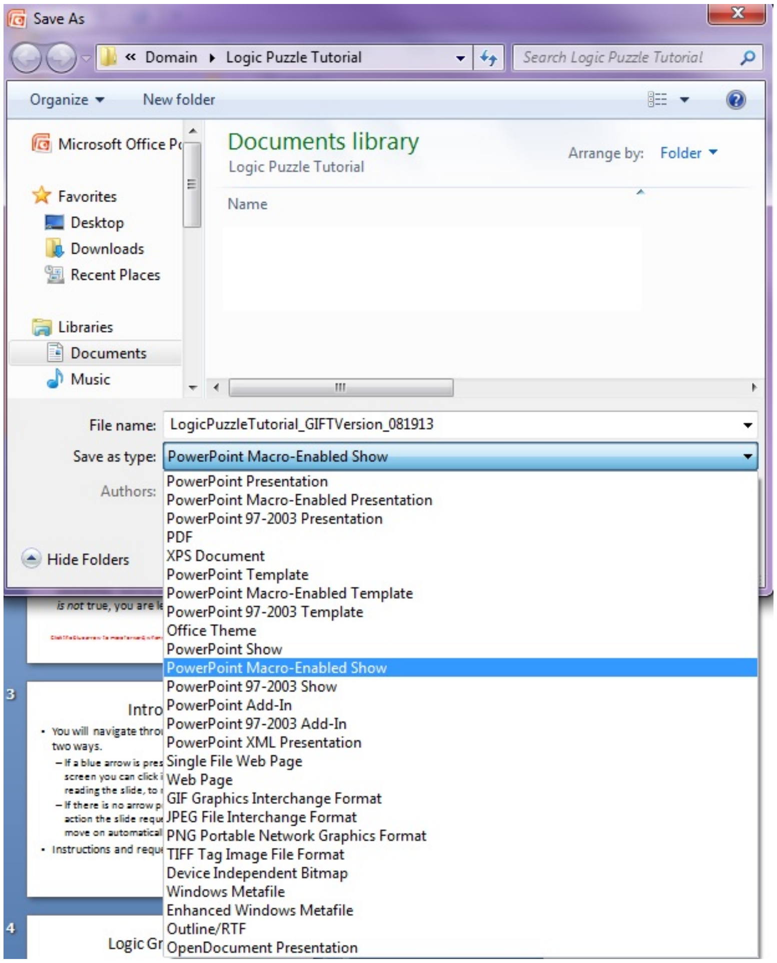
● Present a simple text message
○ Show a file
○ Display a web resource

Message Content:

Thank you for your participation. We will first start with some brief questions that we would like you to answer. Please click continue to proceed...

Figure 2. The same course from Figure 1 in GIFT 2015-1's GAT. Note the improvements to the authoring capabilities, drop down menus, and easy availability of other authoring tools such as the SAS.

The main divisions of GIFT course components have stayed consistent between the versions of the authoring tools. They include the ability to add after action reviews (AAR), surveys, guidance, lesson material, Merrill's branch points and training applications to the course. AAR provide feedback to the learner, however, in order for this to be completed, a DKF needs to have been authored and used. Surveys are created by the course author and can include both multiple choice and short answer components. Only multiple choice questions can be self-graded by the system, but there are many different question types that are available for students to interact with to review their knowledge. Guidance provides information to the students between the different components of the course, and can be a means of providing instructions or brief course content. Lesson Material provides the ability to include PDFs and html as part of the continuous lesson. Merrill's Branch Point refers to the EMAP and authoring different paths that have remediation. Finally, Training Application is an external program that GIFT will be opening and closing for the user such as PowerPoint, VBS2, or TC3Sim (VMedic). In order to use PowerPoint with GIFT, the file itself will need to be saved as a PowerPoint show instead of a normal PowerPoint presentation. This is a fairly easy process. Additionally, if macros or Visual Basic for Applications (VBA) are used, the appropriate PowerPoint show file extension should be utilized. See Figure 3 for how to save a PowerPoint show. The provided example demonstrates saving a file with macros (.ppsm), however for a traditional PowerPoint show without macros the .pps extension would be used. It should also be noted that even if an adaptive DKF is not used, a placeholder DKF (simplest.dkf, which is included with GIFT) should be associated with the PowerPoint show during course authoring.



Save As
« Domain ▸ Logic Puzzle Tutorial
Search Logic Puzzle Tutorial
Organize ▾
New folder
Microsoft Office Pc
Favorites
Desktop
Downloads
Recent Places
Libraries
Documents
Music
Documents library
Logic Puzzle Tutorial
Arrange by: Folder ▾
Name
Authors:
Hide Folders
PowerPoint Presentation
PowerPoint Macro-Enabled Presentation
PowerPoint 97-2003 Presentation
PDF
XPS Document
PowerPoint Template
PowerPoint Macro-Enabled Template
PowerPoint 97-2003 Template
Office Theme
PowerPoint Show
PowerPoint Macro-Enabled Show
PowerPoint 97-2003 Show
PowerPoint Add-In
PowerPoint 97-2003 Add-In
PowerPoint XML Presentation
Single File Web Page
Web Page
GIF Graphics Interchange Format
JPEG File Interchange Format
PNG Portable Network Graphics Format
TIFF Tag Image File Format
Device Independent Bitmap
Windows Metafile
Enhanced Windows Metafile
Outline/RTF
OpenDocument Presentation

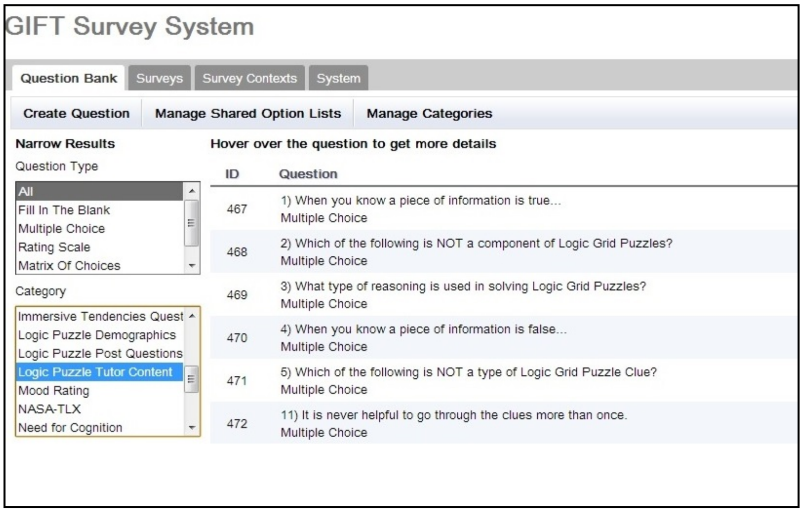
File name: LogicPuzzleTutorial_GIFTVersion_081913

Save as type: PowerPoint Macro-Enabled Show

Figure 3. The appropriate file selection for a PowerPoint Macro-Enabled Show that can be used for GIFT. Other options are to save as a "PowerPoint Show" if it is a normal PowerPoint and no macros are present.

Survey Authoring

The SAS is a powerful tool that can be used by both instructors and researchers. Multiple choice, short answer, fill in the blank and additional question types can be authored by the instructor. The instructor can then combine their questions into a full survey. These questions and surveys are flexible and can be written as quizzes or tests. The surveys can be either used by the student as a way to test his or her knowledge or for grades (however, as previously mentioned in the current versions of GIFT, it is difficult to provide the graded information to the instructor). In order to incorporate surveys into a GIFT course, the instructor will create a survey context that includes all of the surveys that he or she will use during the course, and assign key names for the surveys. The survey context will then be selected when a new course is created using the GAT or CAT, and authors will have access to include their surveys by the assigned key names. Instructors can also author question and survey grading. As mentioned previously, the EMAP allows for adaptive remediation based on performance, however, it is beyond the scope of the current paper. Question banks can be utilized with the EMAP, and they can be authored after creating the survey context in the SAS and associated with specific courses. See Figure 4 for a screenshot of the SAS.



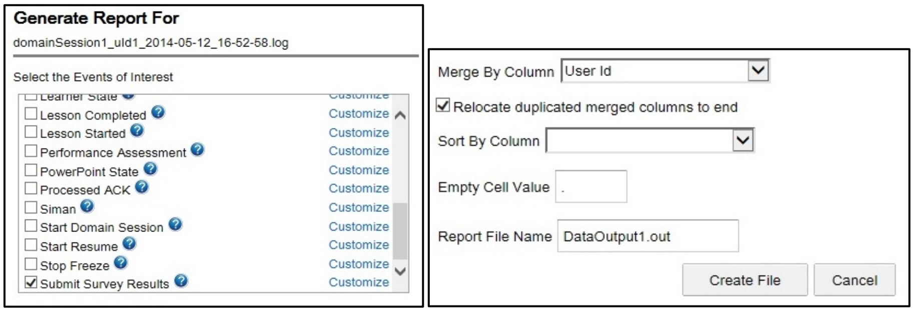
Hover over the question to get more details

ID	Question
467	1) When you know a piece of information is true... Multiple Choice
468	2) Which of the following is NOT a component of Logic Grid Puzzles? Multiple Choice
469	3) What type of reasoning is used in solving Logic Grid Puzzles? Multiple Choice
470	4) When you know a piece of information is false... Multiple Choice
471	5) Which of the following is NOT a type of Logic Grid Puzzle Clue? Multiple Choice
472	11) It is never helpful to go through the clues more than once. Multiple Choice

Figure 4. An example of GIFT's SAS. The tabs at the top of the screen allow the user to switch between the types of items that he or she wants to create

Data Extraction

Data can be extracted using the event reporting tool (ERT). However, as noted, one of the challenges with using GIFT for instruction at the current moment is being able to receive the data from the student. If the student sends the domain log to the instructor or the instructor downloads it off of a lab computer it can be extracted using the ERT. However, depending on how the student logged in to the system, the identifier that the data is merged by may need to be changed. For instance, using the User Id to merge multiple data files may not work properly if two separate students ran the course on different computers as User Id 1. However, in this case, merging by User Name may be more advantageous if each student entered different user names in the system. All of the student entered responses to the GIFT surveys are retained in the domain log files, and the ERT is used to extract them. See Figure 5 for a selection of relevant interfaces for extracting data. On the first ERT screen the instructor selects the log files of interest (more than one file can be selected by holding down "shift" and making selections). In order to be accessed, the log files will need to be in the GIFT\output\logger\message directory. If logs are received from multiple students they can be placed into this directory to be accessed by the instructor's version of GIFT and the ERT. Next, the individual will be prompted to select items they would like in the report (checking Submit Survey Results as demonstrated in Figure 5 will export the survey data). Finally, the individual will be asked how they would like the columns merged and to provide a name for the output. The output will be in the GIFT\output folder, and can be opened in Excel.



Generate Report For
domainSession1_uld1_2014-05-12_16-52-58.log

Select the Events of Interest

□ Learner State — Customize
□ Lesson Completed — Customize
□ Lesson Started — Customize
□ Performance Assessment — Customize
□ PowerPoint State — Customize
□ Processed ACK — Customize
□ Siman — Customize
□ Start Domain Session — Customize
□ Start Resume — Customize
□ Stop Freeze — Customize
☑ Submit Survey Results — Customize

Merge By Column: User Id
☑ Relocate duplicated merged columns to end
Sort By Column:
Empty Cell Value: .
Report File Name: DataOutput1.out

Figure 5. Left: Screenshot of the events of interest that can be selected in the ERT. Right: Screenshot of the merge selection and column sorting options in the ERT.

USING GIFT TO CREATE ASSIGNMENTS

GIFT can be used to create assignments that can be engaged in either as a component of an in class lecture or as part of an online course. These assignments can consist of having students create their own courses in GIFT, creating their own adaptive ITSs using GIFT, creating their own experiments in GIFT, or having them engage in a GIFT course or experiment and generate a report in regard to the data that was collected.

Creating their own Courses

As GIFT is straightforward to use and has a number of different powerful and easy to access tools, it should be able to be used by students to create their own course materials. Instructors may want to assign students to create lessons and assignments that can be used by other students in the course, or that could be used if they were to teach a lessons to others. The students would need to create their PowerPoint materials on their own, and then go to GIFT to enter their survey questions. After all the course materials are gathered, the student can use the CAT or GAT to generate his or her own course. This type of activity can provide good experience for students that hope to teach at a later time, and can assist students in picking out important points from materials.

Creating their own Intelligent Tutoring Systems

A number of graduate level courses in the field of intelligent tutoring systems provide opportunities for their students to create their own ITS. This is a beneficial task, as it results in students thinking through the material that they are planning to teach, generating questions, generating assessments, and generating adaptive feedback. As it has been noted that traditional ITS are tightly coupled with their topics, GIFT offers a great opportunity for students to create their own ITS in areas that they are interested in. If one student wants to create an algebra tutor they can; if another student wants to create a Spanish-language tutor they can. The task of creating an ITS can be given to students of multiple backgrounds such as students in Education, Psychology, and Computer Science.

Collecting Data and Taking on the Researcher Role

There are existing GIFT courses, such as the Logic Puzzle Tutorial, which include a tutorial, assessment questions, and answers. Students can be taught about the research methods process by having them break into small groups and run through the Logic Puzzle Tutorial course themselves. They may also be encouraged to recruit classmates to run through the tutorial. Afterward, they can take on the role of the researcher and extract the performance data and analyze it. A variation of this assignment has been used in a United States Military Academy at West Point, Engineering Psychology Adaptive Tutoring colloquium (Sottilare, Sinatra, Watson, Davis, King, & Matthews, 2014).

Further, students that are engaged in independent studies and creating their own experiments can use GIFT to generate experiments. Documentation exists regarding the advantages of using GIFT to run psychology experiments (Sinatra, 2014), and it has been used for this purpose in the past (Goldberg & Cannon-Bowers, 2013; Sinatra, 2013). GIFT provides beginning researchers with an opportunity to create their experimental materials and then automate the data collection process. This could be especially helpful for undergraduate and graduate students who wish to run multiple participants at once but do not have the number of assistants that could proctor them if it was not an automated process. GIFT's tools and documentation are a great resource for students who will be conducting their own research studies with GIFT.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

GIFT offers many features that are of interest to individuals who are teaching both in-person and online based courses. Further, GIFT can be used by instructors to either create course content, or have their students actively engage with it and create their own courses. The flexibility that GIFT offers makes it a great resource for instructors.

Improvements are continually being made to GIFT, and functionality increases with each GIFT release. Additionally, GIFT will soon be available in the Cloud. The shift to the Cloud should make GIFT easier for both students and instructors to use. However, it also leads to necessary changes and additions to be made such as user roles, permissions, grade books, and ways for instructors to receive the output of their student's courses. In current form, GIFT can be used to present material for supplemental out of class learning, and for assessments. However, if the instructor wants to be able to retrieve the answers that the students provided it is more difficult. In the future, methods for providing student output data to instructors should be improved. One way to do so would be to have an "Email results or data" screen that would allow students to send the data via email directly to their instructor's email address. While it is currently possible for students to provide the information, it is necessary for them to dig through GIFT's folders, and the files are currently unlocked such that they can be edited by the student before submission. If GIFT is to be used for actual quiz or test grades it is recommended that the domain output files be uneditable by students. Perhaps the future incorporation of user roles into GIFT will assist in setting the permissions that are necessary for this to be done.

In its current state GIFT is still a great tool for instructors, and offers many capabilities that will be of use in classes. The provided tools can be used to create a number of different activities that will engage students in their course material. As GIFT is an open-source project, suggestions from instructors and users on how to improve it are always welcome, and will help to shape the future of GIFT.

REFERENCES

Goldberg, B., & Cannon-Bowers, J. (2013, July). Experimentation with the Generalized Intelligent Framework for Tutoring (GIFT): A testbed use case. *In AIED 2013 Workshops Proceedings Volume 7* (p. 27).

Finkelstein, E., & Samsonov, P. (2007). *PowerPoint for Teachers: Dynamic Presentations and Interactive Classroom Projects*. San Francisco, CA: Jossey-Bass Teacher.

Lim, C. (2004). Engaging learning in online learning environments. *TechTrends*, *48*(4), 16 – 23.

Marcovitz, D.M. (2012). *Powerful PowerPoint for Educators: Using Visual Basic for Applications to Make PowerPoint Interactive, 2nd Edition*. Santa Barbara, CA.: Libraries Unlimited.

Sinatra, A.M. (2013, July). Using GIFT to support an empirical study on the impact of the self-reference effect on learning. In *AIED 2013 Workshops Proceedings Volume 7* (p. 80).

Sinatra, A.M. (2014, June). The research psychologist's guide to GIFT. In *Proceedings of the Second Annual GIFT Users Symposium,* (p. 86).

Sinatra, A.M., Sims, V.K. & Sottilare, R.A. (2014). The impact of the need for cognition and self-reference on teaching a deductive reasoning skill. *Army Research Laboratory Technical Report,* ARL-TR-6961.

Sottilare, R., 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.

Sottilare, R. A., Sinatra, A. M., Watson, J., Davis, Z., King, S., & Matthews, M. D. (2014). An evaluation of the Generalized Intelligent Framework for Tutoring (GIFT) from a researcher's or analyst's perspective. *Army Research Laboratory Technical Report.*

ABOUT THE AUTHOR

***Dr. Anne M. Sinatra** is an Adaptive Tutoring Scientist at the Army Research Laboratory's SFC Paul Ray Smith Simulation & Training Technology Center. Her background is in Cognitive and Human Factors Psychology. She has taught Psychology college courses at both introduction and upper-division levels. She conducts adaptive training research as a member of the Learning in Intelligent Tutoring Environments (LITE) Lab and works on the Generalized Intelligent Framework for Tutoring (GIFT) project.*

Theme VI: Dialogue-Based Tutoring

Design of a Domain-Independent, Interactive, Dialogue-based Tutor for Use within the GIFT Framework

Elaine Kelsey[1], Fritz Ray[1], Debbie Brown[1], Robby Robson[1]
[1]Eduworks Corporation, Corvallis, Oregon

INTRODUCTION

Dialogue-based Intelligent Tutoring Systems (ITSs) attempt to emulate human-to-human conversations. Results suggest that they can improve learning outcomes, but they require domain-specific conversational capabilities. To meet these requirements, many such tutors use domain-specific, pre-programmed scripts or templates. These require significant time and effort to produce, thereby limiting the deployment of dialogue-based tutors in educational and training settings.

In this paper we present an alternative method of developing dialogue-based tutors that combines automatically generated domain-specific content with generalized domain-independent dialogue frameworks and the adaptive capabilities of the Generalized Intelligent Framework for Tutoring (GIFT), a flexible, open-source platform developed by the US Army Research Lab that can be used to implement wide a variety of ITS (Sottilare, Brawner, Goldberg, & Holden, 2012; Sottilare & Holden, 2013). The result is a lower-cost method for creating dialogue-based ITS that address specific domains and interact naturally with learners. This paper discusses the architecture, construction, and deployment of these tutors. Topics covered include: an overview of TRADEM ("Tools for the Rapid Automated Development of Expert Models"), taking advantage of GIFT adaptation and sequencing, the JChatScript dialogue engine used to drive interactions; the challenges of automated transformation for dialogue tutors; and semi-automation of dialogue extraction from domain content.

The Goal – Automated Generation of Dialogue-based Tutors

A primary advantage of human-to-human tutoring is the use of adaptive dialogue strategies to scaffold instructional content delivery and to create a supportive, affect-aware environment. Dialogue-based ITSs attempt to emulate these interactions. Results to-date show that these tutoring systems are more effective at improving learning outcomes than those that do not incorporate natural language interaction (Graesser, Conley, & Olney, 2012; Rus, D'Mello, Hu & Graesser, 2013). This may be especially true when there is a difference between the level of material and the level of the student (VanLehn, et al., 2007), which can easily occur when using multiple sources to develop an ITS, as is done in the work reported in this paper.

One of the biggest challenges to scaling up the use of these tutors has been the cost and time involved in generating an effective tutor from an existing corpus of domain knowledge. Most documented instances of successful, interactive, dialogue-based tutors at least partially rely on domain-dependent rules for macro-level topic sequencing, mezzo-level adaptation, and/or micro-level dialogue generation, as defined in Brown, Martin, Ray, and Robson (2014). These tutors are highly effective at emulating human-to-human tutoring within a specific domain, but the addition of each domain requires authors with domain knowledge to create domain-specific rules and content and to possess the skills needed to translate instructional strategies into dialogue-based tutoring techniques. The time and resources required to develop (or update) domain-dependent tutors is a significant barrier to their widespread adoption (Robson & Barr, 2013). Additionally, the associative and social nature of natural discourse in human-to-human tutoring sessions is difficult to emulate with domain-specific tutors since they cannot usually integrate knowledge from a broad range of domains into a single tutoring session. This can undermine the effectiveness of the tutor, especially for learners who interact with the tutor in unanticipated ways.

In Brown, Martin, Ray, and Robson (2014) we described a dialogue-based tutor in which GIFT is used as the adaptation engine for macro-level sequencing of topics within a domain, selection of appropriate instructional strategies, and for maintaining a learner model. This paper describes research into the transformation of a wide range of content inputs into such tutors, with the goal of designing an automated or semi-automated authoring and transformation process that can be applied to any domain of study.

TRADEM and GIFT Integration

Dialogue-based tutors usually require significant transformation of existing source materials and must respond to a wide range of potential user inputs. As such, they are one of the most labor-intensive types of tutors to produce. The TRADEM process (described below in Figure 1) can be applied to generate other types of ITS but is of particular interest for generating dialogue-based tutors.

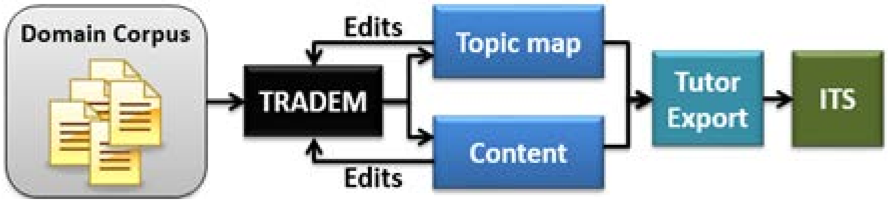


Figure 1. The TRADEM process.

The TRADEM process starts with a corpus of traditional content, representing instruction and knowledge in a target domain. TRADEM then uses natural language processing (NLP), machine learning (ML) and statistical techniques to analyze the corpus. The outputs of this initial process are a dependency map of topics identified within the corpus and a ‘mini-corpus’ of keywords, content and questions associated with each topic. In the next step, the user may modify the topic map by reorganizing topics and relationships, revising and adding content and questions, and inserting additional resources as needed. The contents of each topic are analyzed and reviewed by the user, and portions of the content (referred to as “granules”) are labeled as specific types of instruction (such as facts or examples) and associated with instructional strategies (such as Gagne’s “gain attention”, “present material”, or “elicit performance”). The granules, which may overlap, are tagged with metadata that is ultimately used both to direct the transformation of content into forms that are usable by a conversational tutor and to provide an overall sequence of instruction for a topic. The outputs of these steps are compiled in a JSON representation.

The next step is to define the scope of a course by choosing topics and selecting an export format for a tutor. Formats include a standard zip package, a GIFT PowerPoint package, and a GIFT TRADEM Tutor (T-Tutor) package. For all GIFT exports, TRADEM creates a course package that contains the files required by the GIFT course import and that may be further edited using GIFT authoring tools. The tutor export is automatically configured to support the macro-sequencing of topics within the domain and dynamic selection of instructional strategies within each topic (referred to as mezzo-adaptation in Brown, Martin, Ray, & Robson, 2014) using GIFT's Engine for Management of Adaptive Pedagogy (EMAP). The GIFT course package contains the Course XML file, the EMAP-partitioned segments of instructional content for each topic, the EMAP-encoded metadata for each segment of instructional content, and the course's EMAP-encoded question bank in the form of Survey Authoring System JSON. This approach allows GIFT to act as both a sequencing and assessment engine for a series of customized instructional content segments.

The T-Tutor export further combines each granule with domain-independent micro-adaptive dialogue rules and rules for accessing content from the domain-independent T-Tutor knowledge base. In the future, T-Tutor metadata combined with additional EMAP functionality may allow GIFT to use learner state data to specify which version of micro-adaptive dialogue rules T-Tutor should apply when running a specific granule. In other words, in future versions, the GIFT-T-Tutor integration may be able to recommend micro-adaptive T-Tutor behavior based on a combination of topic sequencing, instructional strategy selection, learner affect, learner performance, and sensor information.

The Process of Content Transformation

As a first step in the process of generating T-Tutor dialogue packages, TRADEM digests domain content into dialogue-sized phrases and outputs that can be used to build JChatScript scripts. It also tags each granule with detailed instructional strategy metadata. The next natural step, for an author or an automated system, is to transform the content into dialogue that fulfills its instructional strategy requirements as indicated in its tags. Using GIFT's "Example strategy" as an illustration, this entails transforming a paragraph of facts into knowledge-based tutoring scripts that follow a prompt, pump, hint, forced-choice dialogue pattern as described in Lehman, D'Mello, Cade, and Person (2012). Other instructional strategies typically require different patterns and transformations.

Regardless of the transformation, it is necessary to have deviations in dialogue to make the interactions seem more natural. To facilitate this, JChatScript allows the user to ask questions of the system, to which it can respond using multiple dialogue scripts. This behavior is an important feature that distinguishes a system that simply presents content in a dialogue fashion from an effective dialogue-based tutor. In particular, the system must respond to several forms of learner inquiries ranging from asking for the definition of a word to asking for facts to questions that engage higher levels of knowledge and understanding. In the prototype version of TRADEM, significant manual effort is involved in creating the corresponding scripts. NLP and ML techniques can be used to automate script creation, with the caveat that the results would be expected to have applicability to all possible scenarios.

Human tutors use diagrams, text, images and other learning aids in addition to and along with dialogue. T-Tutor can display any web-compatible resource ranging from documents and images to HTML5 or

Flash simulations, and the timing of these displays can be synchronized or driven by scripts. However, automatically synchronizing dialogue with resources can be problematic as most content only makes a passing reference to resources and there are many occasions where no reference is made at all. It is possible to generate useful dialogues about a resource if the machine can detect a relationship to specific text passages, but this is difficult to do in the general case and so generating and associating scripts with resources and managing the timing of displays remains a manual step.

Challenges in Automation

The following automation challenges were identified and addressed by the project:

Identification of Discrete Topics from the Target Domain Corpus

The corpus of existing documents for the target domain can include varied levels of pre-existing structure and thematic organization. Different documents may employ different logical frameworks for presenting and sequencing content into sub-topics. There is often significant repetition of topics and information across different documents. To create an overarching model for all content across all documents, an underlying topic structure must be derived from the corpus and individual segments of corpus text must be associated accurately with a best-fit topic. TRADEM suggests the number of topics that optimizes this model. However, the user can change number of topics and adjust the assignment of material to topics. In our experience, the number of topics suggested by TRADEM is usually best.

Labelling of Topics

TRADEM does not extract a set of topic names from a document for several reasons. First, multiple documents are used as input and even if they were organized explicitly by topics (e.g. in tables of contents), there is no guarantee that the labels would agree. Second, even if they were on only one document, the computer-generated topics might differ from the labels in the document because the computer uses semantic analysis rather than domain knowledge to derive them. Third, topics can be split apart and re-combined in a non-transparent way by changing the number of topics that are generated in the first step of the TRADEM process. Therefore, a ‘best-fit’ topic label must be assigned to each topic. The label must be both broad enough to capture the full range of information covered within that topic and specific enough to accurately identify the specific subject of that topic. TRADEM automates this process as described in Robson and Robson (2014).

Mapping of Topic Dependencies

Determining a prerequisite structure for sequencing topics is necessary to produce the course file for GIFT. The data generated by TRADEM includes topic dependencies and prerequisites. The dependency structure is determined by combining several appraisals, including the physical ordering of the corpus texts associated with each topic in the original materials and the degree to which other topics are referenced in corpus texts for any given topic. The TRADEM process for determining topic dependencies is described in Robson, Ray, and Cai (2013).

Associating Topics with Content to form a Mini-Corpus

In the first versions of TRADEM, granules essentially consisted of paragraphs extracted from the input corpus. However, it was quickly discovered that single paragraphs may address multiple topics and that TRADEM needed to handle other structures such as bulleted lists. There is also a need to handle videos, diagrams, and other types of non-text-based content that may not address a single topic. The current version of TRADEM can subdivide structures such as paragraphs into multiple granules and can associate content with multiple topics.

Allocation of Mini-Corpus Topic Content into Instructional Granules and Annotation with Metadata

Within a mini-corpus, content is associated with instructional strategies. Some content may be used (with different transformations) in multiple strategies. TRADEM now allows “many-to-many” associations to be made. In addition, in order for an adaptation engine (i.e. GIFT EMAP) or a dialogue-based tutor to correctly identify the appropriate usage scenarios for a given granule, each granule must be tagged with appropriate metadata. The tags enable T-Tutor to use granules as instructional elements, to respond appropriately to the learner state, and to adopt an appropriate range of micro-adaptive dialogue strategies. In the T-Tutor design, each granule is annotated with a level in Bloom’s taxonomy, a difficulty level, an interactivity level, a presentation order relative to other granules, and its stages in the GIFT EMAP instructional strategy (Goldberg, 2013).

Transformation of Granule Content into Domain-Specific Dialogue Content

Text content must be adapted to flow naturally when incorporated into conversational dialogue. In the case of T-Tutor, the transformed content must also be converted into JChatScript syntax files that can be directly read by the dialogue engine. TRADEM automates the transformation to syntax files, subject to manual editing as needed.

Development of Domain-Independent Dialogue Templates for Instructional Strategies

Certain aspects of dialogue formulation by a human tutor are largely domain-independent. Generation of these other dialogue components is guided by other metadata, e.g. instructional strategy being attempted, learner state, difficulty of the material, and so on. These elements of tutoring dialogues are generic templates that have been developed separately. We anticipate the library of such templates to grow over time.

Integration of Domain-Specific and Domain-Independent Dialogue Components

Dialogue generated by human tutors will generally combine both domain-specific and domain-independent elements in a single dialogue output. To generate this kind of natural dialogue, the transformation individual phrases within a granule are tagged with their potential uses, e.g. as questions, as didactic content, as reflective passages, and so on. The domain-independent dialogue strategies are realized as “plug-ins” that are associated with granules at the phrase level. This gives T-Tutor the ability to select the appropriate dialogue at the right time, whether it is domain-specific or domain-independent.

Generation of Broad Ontologies and General Knowledge to Supplement Domain-Specific Content

Responsiveness to an extremely wide-range of learner queries is a hallmark of human tutors. For example, a learner may apply existing knowledge from another domain or attempt to draw an analogy to the target domain and query the tutor as to the appropriateness of the analogy. Learners may also query the tutor about specific terms and concepts that are not explicitly defined within the target domain corpus. The ability of the T-Tutor to identify when a learner query cannot be addressed through domain-specific materials, and to search for related materials in other domains, is critical to emulating the experience of human-to-human tutoring. If T-Tutor cannot find an appropriate response in the current domain, it can search through other domains for a response. If T-Tutor cannot find a response, it deploys a strategy that amounts to avoiding the question and returning to the topic being tutored. We anticipate that as T-Tutor learns about more domains in the future, it will become increasingly natural in its responses.

Linking Domain-Specific Content with Appropriate Micro-Adaptive Dialogue Rules

While the metadata associated with a specific granule will provide the dialogue engine with some guidance on which conversational rules to apply, the dialogue engine must also be capable of micro-adaptation to changes in learner affect during delivery of a specific granule, such as expressions of surprise, frustration, or boredom. Systems such as GIFT and AutoTutor (Graesser, Weimer-Hastings, Weimer-Hastings & Kreuz, 1999) are instrumented with this capability. Dialogues within a granule are currently labelled as being appropriate for use with specific learner affective states, but incorporating affective response is not yet implemented.

Semi-Automated Processes – Improved Authoring Tools

While a tool capable of fully-automated processing may seem desirable, there are compelling reasons why authors might prefer a semi-automated process:

- Regardless of the quality of the automation, the “garbage-in, garbage-out” principle applies. In an automated process, the quality of the output tutor depends on the quality of the input corpus. Moreover, problems inherent in the original corpus may be exacerbated in the automated output. For example, contradictions and incompatible points of view in the input documents may persist in fully automated output and become more obvious because they are associated with a single topic.

- Topic naming is highly contextual. In the automated process we use, acceptable topic names are often among the top ones produced automatically, but *the* top one is not necessarily the one ranked highest by our algorithms (Robson & Robson, 2014).

- Authors should have the ability to add dialogue elements that reflect their styles and that give the tutor more personality. Over time, the tutor may build up a robust library of domain-independent dialogue elements, but work on each specific domain is likely to be limited to a small number of projects for which authors are required to produce sufficient diversity in dialogues.

- Acronyms, domain-specific terminology, and other important terms may not appear in the original input corpus. An author must add these if they are to be included in the T-Tutor’s vocabulary or explained in a displayed resource or dialogue.

The TRADEM Tutor – Results-to-date

TRADEM includes user interfaces to allow authors to view the automatically-generated content at multiple points in the process and selectively modify it before the next processing step is completed. Examples include revising topic titles, reorganizing content among topics, reviewing and editing metadata, and modifying automatically generated assessment questions.

Automation of authoring has been demonstrated for test cases at all of the points described in this paper, illustrating that fully-automated production of a dialogue-based tutor from target domain source material is possible with variable quality. Although a semi-automated process is needed to produce the highest quality outputs, semi automation still significantly reduces the time required. Ongoing research in TRADEM is focused on improving the quality of the results at each step.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

Fully-automated generation of tutoring packages produces significant time-savings in authoring, but variable results. Key factors in determining the quality of outputs are the degree of transformation required to generate realistic dialogue outputs from excerpts of the original corpus texts, and the extent to which the original texts provide appropriate source materials covering a reasonably broad range of user inputs and queries that may arise in a natural language dialogue exchange. Some script outputs (e.g. facial expressions) cannot currently be derived from the content of domain-specific texts generated through the TRADEM process and must be added manually. Where higher quality results are desired, use of the semi-automated authoring process can provide a good trade-off between speed and quality, and provide a framework for efficiently capturing the edits of subject matter experts who lack specific experience with tutoring or scripting of instructional dialogue.

While the current model theoretically provides for the possibility of generating many different micro-adaptive templates for delivery of a single topic granule (based on learner model data), in practice, adaptation is currently restricted to mezzo-level adaptation of the instructional strategy as described in Brown, Martin, Ray and Robson, 2014. In the future, it may be possible to add metadata to the T-Tutor export package that allows T-Tutor player to make use of GIFT learner model and learner state data in the selection of micro-adaptive rules for the delivery of individual granules. In addition, the dialogue exchanges captured by the T-Tutor player contain potentially useful information about learner state. Analysis of dialogue logs, for the purpose of extracting learner state information for use by GIFT, is a future avenue of research.

ACKNOWLEDGMENTS

TRADEM resulted from a Small Business Innovation Research (SBIR) contract sponsored by the US Army Research Lab. JChatScript is a dialogue engine based on the ChatScript engine (http://chatscript.sourceforge.net) and which was developed in Java™ as part TRADEM.

REFERENCES

Brown, D., Martin, E., Ray, F., & Robson, R. (2014). Using GIFT as an adaptation engine for a dialogue-based tutor. *GIFTSym2*.

Goldberg, B. (2013). GIFT's Engine for Macro/Micro-Adaptive Pedagogy(eM2AP) and Micro-Adaptation Considerations Across a Domain-Independent Architecture: LITE Lab, Army Research Laborartory – Human Research & Engineering Directorate (HRED).

Graesser, A. C., Conley, M. W., & Olney, A. (2012). Intelligent tutoring systems. In S. Graham, & K. Harris (Eds.), APA Educational Psychology Handbook: Vol. 3. Applications to Learning and Teaching (pp. 451-473). Washington, DC: American Psychological Association.

Graesser, A., Wiemer-Hastings, K., Wiemer-Hastings, P., & Kreuz, R. (1999). AutoTutor: A simulation of a human tutor. *Cognitive Systems Research*, *1* (1), 35-51.

Lehman B., D'Mello S., Cade W., and Person N. (2012). How do they do it? Investigating dialogue moves within dialogue modes in expert human tutoring. *Intelligent Tutoring Systems, 11th International Conference*, 557-563.

Robson, R., & Barr, A. (2013). Lowering the Barrier to Adoption of Intelligent Tutoring Systems through Standardization. In Design Recommendations for Intelligent Tutoring Systems, Vol 1, Sottilare, R., Graesser, A., Hu, X., & Holden, H. (Eds.), US Army Research Lab, 7 – 13.

Robson, R., Ray, F., & Cai, Z. (2013). Transforming Content into Dialogue-based Intelligent Tutors. In The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC) (Vol. 2013, No. 1). National Training Systems Association.

Robson, E. & Robson, R. (2014). Automated Content Alignment for Adaptive Personalized Learning. *Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2014*, Orlando, Florida, National Training and Simulation Association.

Rus, V., D'Mello, S.K., Hu, X., & Graesser, A.C. (2013). Recent advances in intelligent tutoring systems with conversational dialogue. *AI Magazine*, 34, 42-54.

Sottilare, R., 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.

Sottilare, R. A., & Holden, H. K. (2013). *Recommendations for Authoring, Instructional Strategies and Analysis for Intelligent Tutoring Systems (ITS): Towards the Development of a Generalized Intelligent Framework for Tutoring (GIFT).* Paper presented at the Artificial Intelligence in Education 2013, Memphis, TN.VanLehn, K., Graesser, A. C., Jackson, G. T., Jordan, P., Olney, A., & Rosé, C. P. (2007). When are tutorial dialogues more effective than reading?. Cognitive Science, 31(1), 3-62.

ABOUT THE AUTHORS

Ms. Elaine Kelsey *is a software engineer at Eduworks where she focuses on development of conversational dialogue agents for intelligent tutoring systems, natural language processing and semantic analysis.*

Mr. Ed "Fritz" Ray *leads the Eduworks software engineering team. He has architected and developed applications in areas ranging from educational digital libraries to semantic analysis, semantic search, competency management, and patent analysis.*

Ms. Debbie Brown *is the chief learning technologist at Eduworks and has served as a project manager, learning technologist, and researcher for multiple NSF and DOD projects.*

Dr. Robby Robson *is CEO and Chief Scientist of Eduworks and has been engaged in researching and developing learning technologies for almost twenty years, with contributions in the areas of learning management systems, reusability, digital libraries, standards, and intelligent tutoring systems.*

CSAL AutoTutor: Integrating Rich Media with AutoTutor

Zhiqiang Cai, Arthur C. Graesser, Xiangen Hu and Benjamin D. Nye
University of Memphis

INTRODUCTION

AutoTutor started as an intelligent tutoring system that teaches conceptual sciences by holding natural language conversations with learners (Graesser, Wiemer-Hastings, Wiemer-Hastings, & Kreuz, 1999). Since the 1990s, several AutoTutor systems have been developed, including the earlier ones that teach physics, computer literacy and the later ones that teach critical thinking skills (Forsyth, Graesser, Pavlik, Cai, Butler, Halpern, & Millis, 2013), adult literacy (CSAL AutoTutor), and others. AutoTutor Conversation Engine (ACE) is the first external tutoring service integrated into releases of GIFT (Sottilare, Holden, Goldberg, & Brawner, 2013; Nye, Hu, Graesser, & Cai, 2014). In GIFT, AutoTutor is responsible for handling intelligent natural language conversations (Sottilare, 2014).

GAMETE (Game-based Architecture for Mentor-Enhanced Training Environments) provided a practical example of linking ACE to game-based learning environments (Engimann, Santarelli, Zachary, Hu, Cai, Mall & Goldberg, 2014). GAMETE sets up links between AutoTutor and a game environment, TC3sim (Tactical Combat Casualty Care Simulation). Learners' performance in TC3sim is used to determine what AutoTutor conversation to invoke. Learners' performance in the AutoTutor conversation is used to decide the next step of the game. This kind of integration is "interruptive", in the sense that any time a conversation starts, the game interaction is interrupted. A learner does not interact with the game until the conversation ends.

In order to create a non-interruptive learning environment, a deeper communication channel between ACE and other interactive elements in the learning environment has to be available so that ACE not only knows the interactions on other elements but also is able to suggest changes to other elements. The red arrow in Figure 1 shows the communication channel between media elements and ACE. This paper talks about this channel in detail. As an example, we use CSAL AutoTutor to illustrate how ACE could be seamlessly integrated.

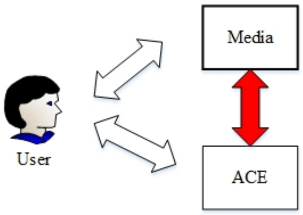
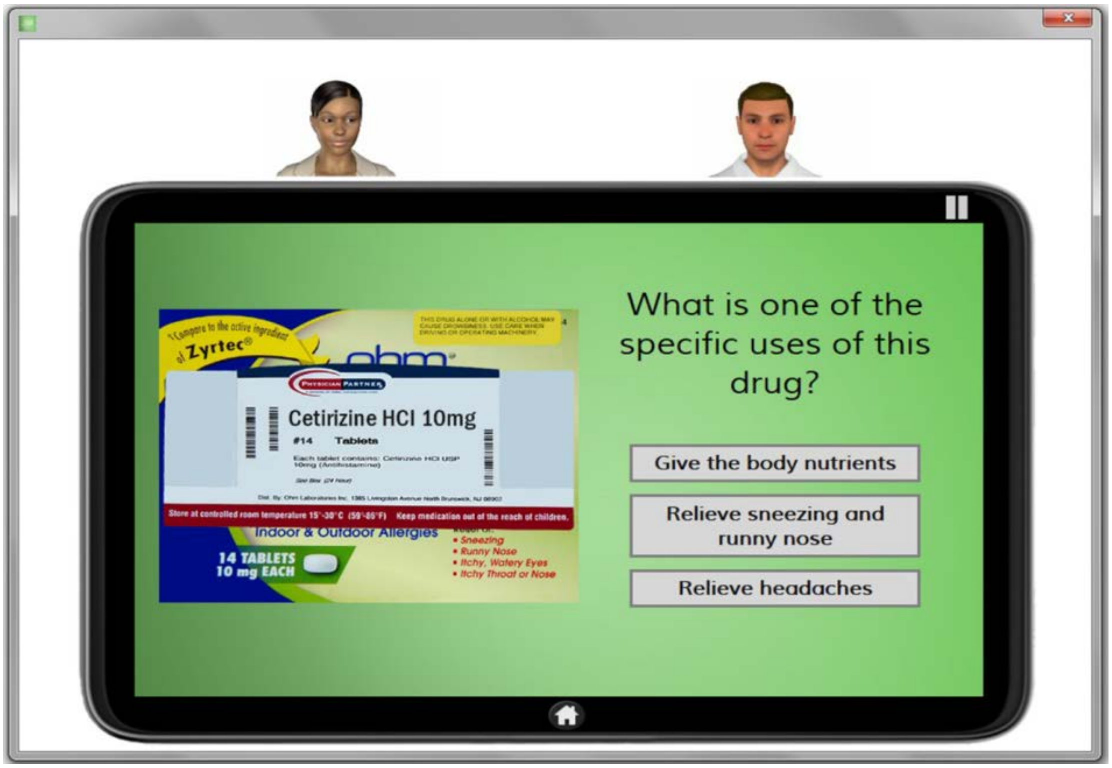


Figure 1. User Interaction with Media and ACE.

ARCHITECTURE OF CSAL AUTOTUTOR

The Center for Study of Adult Literacy (CSAL) is a national research center funded by the Institute of Education Sciences. CSAL AutoTutor is a learning system that is designed to help adult learners (3rd to 8th grade levels) improve their reading ability. One critical constraint in developing this system is that many adult learners have limited ability in typing their own words into the system. Therefore, the system has to use more mouse actions as an input method. Although voice might be a better input option, the cost is too expensive on both development and usage: it is currently infeasible to distribute speech-quality microphones across adult literacy centers at scale. For this reason, CSAL AutoTutor does not use voice input. Instead, it starts with non-verbal input and gradually brings learners to the higher levels that require more and more typed verbal input.

Figure 2 shows the interface of CSAL AutoTutor. It is a simple HTML page that contains two iFrames, an agent iFrame and a media iFrame. The agent iFrame has two animated agents. The left one, Cristina, plays a role as a teacher and the right one, Jordan, plays a role as a peer student. The agents were built using MediaSemantics Character Builder. The media iFrame is used to load HTML pages that are created as the learning content. A content page may contain static elements, such as texts and images. It may also contain interactive elements, such as videos, buttons, menus, drag and drop objects, highlightable objects, text input box, etc. During the learning process, ACE needs to know what content page is loaded and what interaction is done on an element. ACE also needs to be able to send commands to the content pages to perform actions, such as, lock/unlock, enable/disable, highlighting, and so on.



What is one of the specific uses of this drug?

Compare to the active ingredient of Zyrtec®
THIS DRUG ALONE OR WITH ALCOHOL MAY CAUSE DROWSINESS. USE CARE WHEN DRIVING OR OPERATING MACHINERY
ohm
PHYSICIAN PARTNERS
Cetirizine HCl 10mg
#14 Tablets
Each tablet contains: Cetirizine HCl USP 10mg (Antihistamine)
Store at controlled room temperature 15°-30°C (59°-86°F) Keep medication out of the reach of children.
Indoor & Outdoor Allergies
14 TABLETS 10 mg EACH
• Sneezing
• Runny Nose
• Itchy, Watery Eyes
• Itchy Throat or Nose
Give the body nutrients
Relieve sneezing and runny nose
Relieve headaches

Figure 2. CSAL AutoTutor Interface

Figure 3 shows the architecture of CSAL AutoTutor. The main page of CSAL is responsible for the communication among the agent, media and ACE. When a CSAL AutoTutor session starts, CSAL AutoTutor loads to ACE a script of a lesson selected by a user. ACE then sends the first set of commands to the CSAL AutoTutor main page. The main page interprets and executes each command. An executable command either invokes an agent speech or causes changes to the loaded media page. After all commands are executed and the agent speeches are finished, the main page checks if there are any interactions available on the loaded media page. If no interaction is available, the main page calls ACE and gets the next set of commands. If interactions are available, the main page waits for the user's input. Once an input is received, the main page sends the input to ACE for interpretation and gets back a new set of commands.

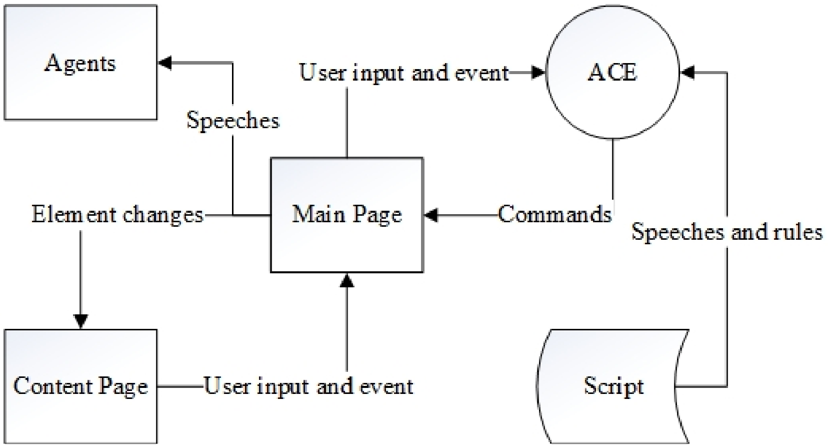


Figure 3 CSAL AutoTutor Architecture

Setting up the communication is challenging. On one hand, content page developers need freedom to create arbitrarily complex changeable and interactive elements. On the other hand, ACE needs to be aware of the outcome of interactions and possible ways to change the elements on the page. The outcome of interactions decides what the next step is. The possible changes are used to construct ACE commands. These are specified as conversation rules in AutoTutor scripts.

AUTOTUTOR SCRIPT

An AutoTutor script contains two major parts. One part is a set of conversation elements, including agent speeches and target answers for matching user responses. The speech elements could be a rigid conversation between computer agents. It could also be a complex AutoTutor tutoring dialogue based on Expectation-Misconception Tailored interactions (EMT; Graesser, Li, & Forsyth, 2014). A typical EMT package contains a main question, a set of expectations and misconceptions, hint and prompt questions, and answers to each question. Each question may correspond to multiple types of answers, such as "Good", "Bad", "Partial", etc. The answers are used to match learners' input. An answer may also be spoken by a computer agent. "Good" answers are usually spoken by the teacher agent and "Bad" answers are usually spoken by the peer student agent.

Another major part in an AutoTutor script is a set of production rules. An AutoTutor rule contains a condition part and an action part. The condition part contains seven fields that are used for rule selection.

Condition

- Status – "Status" is used to track where the learning session is. For any given status, there could be multiple rules. The status for the first rule to execute is "Start". In each rule, there usually contains an action for updating the status. ACE keeps track of the status and searches rules within those that match the status.

- Response – "Response" is the text input category, such as "Good", "Bad", "Partial", "Irrelevant", "Blank", etc. Script authors can create customized categories. The categories are determined by matching the text input to stored targets using LSA (Latent Semantic Analysis; Landauer, McNamara, Dennis, & Kintsch, 2007) and Regular Expressions.

- Event – "Event" is a label that represents what happened in the loaded media page. For example, "PageLoaded" may indicate that a new page has just been loaded; "Correct" may indicate that a user selected a correct answer; "Paused" may indicate that a video is paused by the user. Media pages are responsible for generating the labels. Obviously, media page developers and the script authors need to share the label lists and agree on what each label indicates.

- HasItem – "HasItem" is used to indicate whether or not a specific script item is available. This is usually used to continue or terminate a loop. For example, the rules can be set up to go back repeatedly to a "GetHint" Status until "HasItem" becomes "False".

- Priority – "Priority" is used to prioritize rule selection. Because of this, other rule conditions do not have to be exclusive. For example, a "Good" answer may imply a "Partial" answer. However, a higher priority can be assigned to "Good" answer, so that when "Good" answer is matched, "Partial" answer will not be considered. One important use of "Priority" is to assign the lowest priority to an "Otherwise" rule. The "Otherwise" can match anything, except that it will be considered only when no other rule is selected. The "Otherwise" rule can keep the system from freezing.

- MaxVisit – "MaxVisit" is another way to terminate a loop. It sets up the maximum number of times a rule may be selected.

- Frequency – "Frequency" is used to randomly select from multiple matched rules with given probability. For example, if rule A and rule B both match other conditions and have the same priority, the frequency of the two will be used to compute the probability by the formula:

$$\boldsymbol{Probability\ of\ A = \frac{Frequency\ of\ A}{Frequency\ of\ A + Frequency\ of\ B}}$$

Action Set

An action set contains a sequence of commands that are used for further rule search or for client execution. Each command contains 4 fields.

- ID – "ID" specifies the order of the actions. Actions are sorted by ID with increasing order and then executed one by one. IDs are integers but do not have to be consecutive.
- Agent – "Agent" can be the ID of any of the computer agents or "System".
- Act – "Act" is the type of action. There are a set of reserved action types in ACE. However, the action type can be arbitrarily created by script authors. Script authors share action lists with the interface developer and make sure the actions will be correctly interpreted by the main page.
- Data – "Data" is information needed by the given type of action. It can be blank if no data is needed.

COMMUNICATION BETWEEN ACE AND MEDIA PAGES

In CSAL AutoTutor, the main page talks to content pages by JavaScript functions. When anything happens on a loaded content page, the content page may send a message to the main page by calling a main page function. The main page may call a content page function to get specific information or make changes to the content page. The main page calls ACE when a message is received from the loaded content page that needs ACE to process.

Let us use Figure 2 as an example to help illustrate this communication mechanism. The content page on Figure 2 shows a question, "What is one of the specific uses of this drug?" There are three answer options for a user to choose: A) "Give the body nutrients"; B) "Relieve sneezing and running nose"; and C) "Relieve headaches". The text on the drug instructions shows that B) is the correct answer.

Table 1. Two example rules in an AutoTutor Script.

World Event	Actions		
	Agent	Act	Data
Correct	Cristina	Speak	Great job!
	System	MediaMessage	Next Page
	System	SetStatus	NewPage
	System	WaitForPageLoading	
Incorrect	Cristina	Speak	That is not right. Try again.
	System	MediaMessage	Second Try
	System	SetStatus	SecondTry
	System	WaitForUser	20

When a user clicks on a button to choose an answer, a JavaScript function sends the main page a message of “Correct” or “Incorrect”, according to what the user chooses. The main page then calls ACE with this message. When ACE receives this message, it searches in the script for a rule at this given state with the world event as “Correct” or “Incorrect”. Table 1 shows two sets of actions corresponding to “Correct” and “Incorrect”, respectively.

If the user’s answer is correct, then the first 4 actions in Table 1 will be chosen by ACE. ACE executes the “SetStatus” action and updates the cached status to “NextPage”. It then sends three actions to the main page. The main page first sends “Great job!” to the Agent page for Cristina to speak. After Cristina finishes speaking, the main page sends a message to the content page to load the next page. This implies that there is a JavaScript function on the content page that can load the next page. Then the system waits for the loading message from the content page. The loading message could be “PageLoaded” when the next page is loaded, or “NoMorePage” if no more next page exists. Of course, two separate rules are needed to handle these two situations. A similar process occurs for “Incorrect”.

In CSAL, a list of common functions is used across all content pages. We list them below and believe they are useful for content developers who use the CSAL architecture.

- Ready – Sends a “PageLoaded” message to the main page when the page is fully loaded.

- Lock – Locks the interactive components on the content page. This is usually used when the agents are speaking.
- Unlock – Unlocks the interactive components for the user to interact.
- ShowItem(ItemName) – Shows hidden items or adds animations to the item (e.g., highlighting).
- NextPage – Loads the next page. If no next page exists, this function sends a "NoMorePage" message to the main page.
- GetScore(Name) – Returns the user's score on the content page for the specified score name.
- GetHistory – Returns a string that represents what has happened on the content page up to the time the function is called.
- MoveToHistory(History) – This function uses the string returned from the GetHistory function to restore the content page to the specific point. This function is used for system recovery. That is, if a user pauses at a certain point and logs on to the system again, the system can load the last content page and move to the point where the user paused.
- GetProgress – This function returns an estimated progress value. The value should indicate the percentage of tasks the user has completed for a given session when the specific content page is reached.

CONSIDERATIONS FOR CSAL AUTOTUTOR AUTHORING

CSAL AutoTutor has a simple architecture that allows authors to integrate natural language conversations with interactive content pages. To create a new CSAL AutoTutor style lesson, an author needs to represent the lesson materials by a set of HTML pages, with appropriate JavaScript functions. The author also needs to author an AutoTutor script in which, in addition to agent speech, a set of production rules are required. Authoring JavaScript functions and setting up production rules are the most challenging parts for domain experts who do not have enough knowledge in programming.

A template based online authoring environment should help domain experts to easily create eye-catching lessons, if enough templates are provided. The authoring tool should have a continuously increasing library of lesson templates for a domain expert author to choose. Once a template is chosen the author's task is reduced to changing the static contents, such as texts, images and agent speech.

The goal of this paper is to illustrate how ACE can be fully integrated with learning materials. We believe this can be extended to the communication between ACE and other GIFT modules. GIFT modules may use similar mechanisms to communicate with ACE, so that information from those modules can be used in ACE for rule constructions and selections, and ACE actions can be sent to those modules for execution.

REFERENCES

Engimann, J., Santarelli, T., Zachary, W., Hu, X., Cai, Z., Mall, H., & Goldberg, B. (2014). Game-based Architecture for Mentor-Enhanced Training Environments (GAMETE). Proceedings of the 2nd Annual GIFT Users Symposium.

Forsyth, C.M., Graesser,A.C. Pavlik, P., Cai, Z., Butler,H., Halpern, D.F., & Millis, K.(2013). OperationARIES! methods, mystery and mixed models: Discourse features predict affect in a serious game. *Journal of Educational Data Mining*, 5, 147-189.

Graesser, A. C., Wiemer-Hastings, K., Wiemer-Hastings, P., & Kreuz, R. J. (1999). AutoTutor: A simulation of a human tutor. *Cognitive Systems Research*, 1(1), 35-51.

Graesser, A. C., Li, H., & Forsyth, C. (2014). Learning by communicating in natural language with conversational agents. *Current Directions in Psychological Science*, 23, 374-380.

Landauer, T. K., McNamara, D. S., Dennis, S., and Kintsch, W. (Eds.) (2007), Handbook of Latent Semantic Analysis. Mahwah, NJ, US: Lawrence Erlbaum Associates Publishers.

Nye, B. D., Hu, X., Graesser, A. C., & Cai, Z. (2014). AutoTutor in the Cloud: A Service-oriented Paradigm for an Interoperable Natural-language ITS. *Journal of Advanced Distributed Learning Technology*, 2(6), 49-63.

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.

Sottilare, R. (2014). Examining Opportunities to Reduce the Time and Skill for Authoring Adaptive Intelligent Tutoring Systems. Proceedings of *the 2nd Annual GIFT Users Symposium*.

ABOUT THE AUTHORS

***Zhiqiang Cai** is a research assistant professor at the University of Memphis Institute for Intelligent Systems. He has a M Sc. degree in computational mathematics from Huazhong University of Science and Technology. His research interests are natural language processing, algorithm design, and software development for tutoring systems. He is the chief software designer and developer of ACE (AutoTutor Conversation Engine), ASAT (AutoTutor Script Authoring Tool), QUAID, and Coh-Metrix. He has previously been an associate professor at Huazhong University of Science and Technology (1994-2001), Sudan University of Science and Technology (visiting, 1996-2000), and the University of Paris VI (visiting, 1995).*

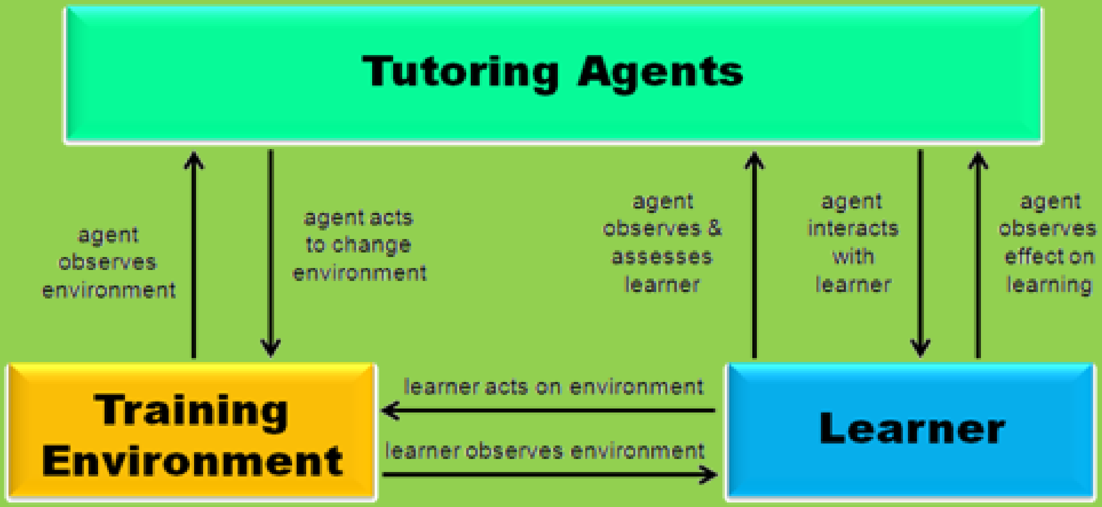
***Dr. Arthur C. Graesser** is a professor in the Department of Psychology and Institute for Intelligent Systems at the University of Memphis, whose primary research interests span cognitive science, discourse processing, and the learning sciences. More specific interests include knowledge representation, question asking and answering, tutoring, text comprehension, inference generation, conversation, reading, education, memory, artificial intelligence, and human-computer interaction. Dr. Graesser received his Ph.D. in psychology from the University of California at San Diego. He has served as editor of the Discourse Processes and the Journal of Educational Psychology. He is past president of the Society for Text and Discourse and Artificial Intelligence in Education. He has published over 600 papers in journals, books, and conference proceedings, written 3 books, and edited 14 books. He and his colleagues have built and tested dozens of cutting-edge learning and discourse technologies, including AutoTutor, Operation ARIES!, Coh-Metrix, and Question Understanding Aid (QUAID).*

***Dr. Xiangen Hu** is a professor in the Department of Psychology at The University of Memphis, with a secondary appointment in Engineering, a senior researcher at the Institute for Intelligent Systems (IIS), and a visiting professor at Central China Normal University (CCNU). Dr. Hu received his Ph.D. in Cognitive Sciences (1993) from the University of California, Irvine. Currently, Dr. Hu is the director of cognitive psychology at the University of Memphis, the Director of Advanced Distributed Learning (ADL) center for Intelligent Tutoring Systems (ITS) Research & Development, and senior researcher in the Chinese Ministry of Education's Key Laboratory of Adolescent Cyberpsychology and Behavior. Dr. Hu's research areas include mathematical psychology, research* *design and statistics, cognitive psychology, knowledge representation, intelligent tutoring, and advanced distributed learning.*

***Dr. Benjamin D. Nye** is a research assistant professor at the University of Memphis Institute for Intelligent Systems (IIS). His research interests include modular intelligent tutoring system designs, modeling social learning and memes, cognitive agents, and educational tools for the developing world. He received his Ph.D. in Systems Engineering from the University of Pennsylvania in 2011. Ben is currently leading work on the Sharable Knowledge Objects (SKO) framework, a service-oriented architecture for AutoTutor. He is also researching and data mining a large corpus of human-to-human online tutoring dialogues, as part of the ADL Personalized Assistant for Learning (PAL) project. Ben's major research interest is to identify barriers and solutions to development and adoption of ITS so that they can reach larger numbers of learners, which has traditionally been a major roadblock for these highly-effective interventions.*

Proceedings of the Third 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.


Tutoring Agents
agent observes environment
agent acts to change environment
agent observes & assesses learner
agent interacts with learner
agent observes effect on learning
Training Environment
learner acts on environment
learner observes environment
Learner

The GIFT Users Symposia were started 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 Editors:

Dr. Robert Sottilare *leads adaptive training research within US Army Research Laboratory's Learning in Intelligent Tutoring Environments (LITE) Lab in Orlando Florida. He is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT).*

Dr. Anne M. Sinatra *leads team adaptive training research within US Army Research Laboratory's Learning in Intelligent Tutoring Environments (LITE) Lab in Orlando Florida. Her research interests include human factors and cognitive psychology.*

Part of the Adaptive Tutoring Series

9 7 8 - 0 - 9 8 9 3 9 2 3 - 8 - 9