Toward a Modular Reinforcement Learning Framework for Tutorial Planning in GIFT

Jonathan Rowe¹, Bradford Mott¹, James Lester¹, Bob Pokorny², Wilbur Peng², and Benjamin Goldberg³
North Carolina State University¹, Intelligent Automation, Inc.², U.S. Army Research Laboratory²

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 (McAlindien, 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, 2015). 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 IN SIMULATION-BASED TRAINING ENVIRONMENTS

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, 2011). 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 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 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

![Figure 1. UrbanSim simulation-based training environment](image)
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 \( \pi \), 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, \( M = \{ M_i \}_1^N \), where each \( M_i = (S_i, A_i, P_i, R_i) \), corresponding to a sub-problem in the composite reinforcement learning task. Each agent \( M_i \) has its own state sub-space \( S_i \), action set \( A_i \), probabilistic state transition model \( P_i \), and reward model \( R_i \). The solution to a modular reinforcement learning problem is a set of \( N \) policies, \( \pi^* = \{ \pi^*_i \}_1^N \), where \( \pi_i \) is the optimal policy for the constituent MDP \( M_i \). Whenever two policies \( \pi_i \) and \( \pi_j \) 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, \( R_s \), 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:

1. **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.

2. **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.

3. **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 FOR DATA-DRIVEN TUTORIAL PLANNING

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 intelligent tutoring systems 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: extended 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


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.