Toward Simulated Students for Reinforcement Learning-Driven Tutorial Planning in GIFT

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

INTRODUCTION

A critical feature of intelligent tutoring systems (ITSs) is their capacity to guide and scaffold student learning. Tutorial planners leverage contextual information to determine how pedagogical feedback, hints, and prompts should be tailored to learners at run-time (Woolf, 2008). Recent years have witnessed growing interest in applying reinforcement learning (RL) to devise tutorial planners. RL provides a data-driven framework for creating tutorial planners from observations of student behavior and learning outcomes. RL techniques introduce the potential for ITSs that can automatically refine and improve their pedagogical methods over time. RL methods account for the inherent uncertainty in how learners respond to different types of tutorial strategies and tactics, and they produce models that can be automatically induced to optimize measures of student learning. Recent work on reinforcement learning-based tutorial planning has outlined a path for devising pedagogical models across a broad range of learning environments and educational domains (Chi, VanLehn, Litman, & Jordan, 2011; Rowe & Lester, 2015; Williams et al., 2016).

An important challenge for RL-based intelligent tutoring systems is the availability of sufficient data to train pedagogical decision-making models. Advanced learning technologies, such as simulations (Mislevy et al., 2014) and digital games (Rowe & Lester, 2015; Shute, Ventura, & Kim, 2013) often present vast action and state spaces, which raise tractability issues for computational models of tutorial planning. Several projects have trained RL-based intelligent tutors using data from human students, but these systems typically rely upon highly constrained state representations and action sets. An alternate approach is leveraging synthetic data generated by simulated students, which can provide large volumes of training data for RL systems. By simulating students' learning and behavior processes, it is possible to generate effectively unlimited synthetic data for training RL-based tutorial planners. However, this approach presents its own set of challenges, including questions about model granularity, validity, complexity, and efficiency.

In this paper, we survey different approaches for creating simulated students and examine their potential for training RL-based intelligent tutors. We describe steps we are taking to leverage simulated students to induce a modular RL-based tutorial planner for counterinsurgency (COIN) training in GIFT. Specifically, we are investigating the design of simulated students for two complementary COIN training environments: UrbanSim and Urbansim Primer. We discuss major considerations in the design of simulated students for our domain, and we conclude with a discussion of potential enhancements to GIFT that would support the creation of intelligent tutoring systems from simulated student data.

REINFORCEMENT LEARNING-DRIVEN TUTORIAL PLANNING

Reinforcement learning refers to a family of machine learning tasks that induce software control policies for sequential decision-making under uncertainty with delayed rewards (Sutton & Barto, 1998). In classical reinforcement learning, an agent seeks to learn a policy for selecting actions in an uncertain environment in order to accomplish a goal. The environment is characterized by a set of states and a probabilistic model describing transitions between those states. The agent is capable of observing the environment's state and using its observations to guide decisions about which actions to perform. The agent's task is to utilize the reward signal in order to learn a policy that maps observed states to actions and maximizes its total accumulated reward.

Reinforcement learning problems are typically formalized using Markov decision processes (MDPs). An MDP is formally defined by a tuple $M = (S, A, P, R)$, where S is the set of environment *states*; A represents the set of *actions* that the agent can perform; P is the *state transition model*, P: $\{S \times A \times S\} \rightarrow [0, 1]$, which specifies the probability of transitioning to state $s_{t+1} \in S$ after performing action $a_t \in A(s_t)$ in state $s_t \in S$ at time step t; and R is the *reward model*, $R: \{S \times A \times S\} \rightarrow \Re$, which specifies the expected scalar reward $r_t \in \Re$ associated with performing action a_t in state s_t and transitioning to state s_{t+1}. The solution to an MDP is an optimal policy, $\pi^*(s_t) \rightarrow A$, that maps states to actions and yields maximum expected reward for the agent. There are a number of algorithms for solving MDPs under different conditions, including both on-line and off-line contexts.

Reinforcement Learning in Intelligent Tutoring Systems

In our work, we formalize data-driven tutorial planning as a modular reinforcement learning task (Fig. 1). Modular reinforcement learning is a multi-goal extension of classical reinforcement learning that divides a decision-making problem into multiple concurrent sub-problems, each modeled as its own MDP, and machine learning individual policies to solve each sub-problem. Formally, modular reinforcement learning tasks are defined in terms of N concurrent MDPs, $M = \{M_i\}_1^N$, where each $M_i = (S_i, A_i, P_i, R_i)$, corresponding to a sub-problem in the composite reinforcement learning task. Each agent M_i has its own state sub-space S_i, action set A_i, probabilistic state transition model P_i, and reward model R_i. The solution to a modular reinforcement learning problem is a set of N policies, $\pi^* = \{\pi_i^*\}_1^N$, where π_i^* is the optimal policy for the constituent MDP M_i. Whenever two policies π_i and π_j with $i \neq j$ recommend different actions in the same state, an arbitration procedure must be applied. Standard reinforcement learning algorithms can be used to compute solutions for the constituent MDPs. In cases where conflicts occur between concurrent policies for multiple sub-problems, arbitration procedures are employed. Because models for each sub-problem are individually learned, they need only consider those state features, actions and goals that are relevant to the sub-problem.

By decomposing tutorial planning into multiple sub-problems, we can reduce the complexity of reinforcement learning by reframing the task in terms of several smaller, concurrent Markov decision processes, which are solved using modular reinforcement learning methods. To perform this decomposition, we employ the concept of an adaptable event sequence (AES), an abstraction for a recurring series of one or more instructionally related events that, once triggered, can unfold in several different ways within a learning environment. To illustrate the concept of an AES, consider the task of selecting an instructional strategy to deploy after a student has made an influential decision in a simulation-based learning environment. In our project, the tutorial planner selects among four possible instructional strategies: 1) providing single-topic coaching on a concept that the learner has not yet mastered; 2)

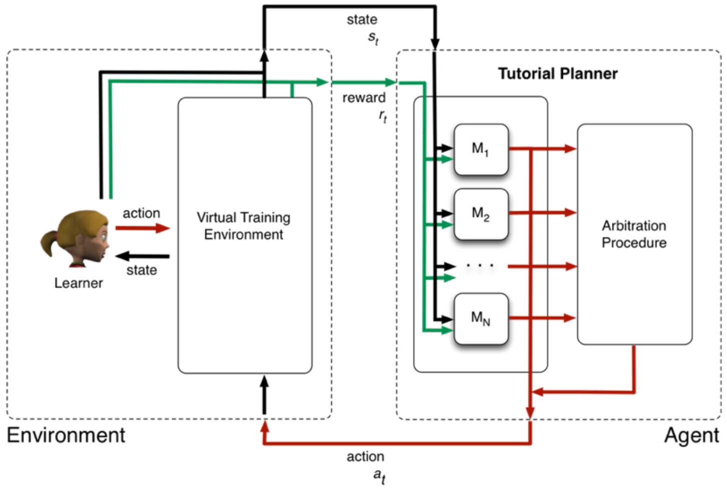


Fig. 1. Modular reinforcement learning framework for tutorial planning.

providing a multi-topic review on several concepts that the learner has had opportunity to practice; 3) providing feedback on unproductive learning behaviors by the student; or 4) not intervening at all. Each of these four responses is an alternate instructional strategy, and they should be varyingly deployed based on the learner's performance and the state of the training environment. Decisions about what type of instructional strategy to utilize are likely to occur multiple times over the course of a student's learning interaction. Because this tutorial sequence can unfold in one of several valid ways, we refer to it as adaptable, or in other words, it is an adaptable event sequence (AES).

Leveraging the concept of an AES, tutorial planning can be cast as a collection of sequential decision-making problems about scaffolding learning within a virtual training environment. Each AES is modeled as a distinct Markov decision process, M_i. For each AES, every occurrence of the event sequence corresponds to a decision point for M_i. The set of possible scaffolding options for the AES is modeled by an action set, A_i. A particular state representation, S_i, is selected for each AES. State encodes the learner's state and history, as well as the learning environment's state. A state transition model P_i encodes the probability of transitioning between two specific states during successive decision points for the AES. Rewards, R_i, can be calculated from formative or summative assessments of student learning, such as a post-test. Reward is the metric that reinforcement learning is designed to optimize.

To induce optimal policies that solve the MDPs, we can utilize two possible approaches. The first approach is to collect training data from human learners by deploying a tutorial planner that selects actions randomly, in effect sampling the space of tutorial policies and rewards (Chi, VanLehn, Litman, & Jordan, 2011; Rowe & Lester, 2015). Leveraging the mapping between AESs and MDPs, and a training corpus of random tutorial decision data, we can employ model-based reinforcement learning techniques to induce policies for tutorial planning. Typically, dynamic programming methods (e.g., value iteration) are utilized to compute solution policies off-line for each MDP using maximum likelihood estimates of the state transition model and reward model computed from the training corpus (Chi, VanLehn, Litman, & Jordan, 2011; Sutton & Barto, 1998).

An alternate approach is to devise a simulated student that generates synthetic training episodes for on-line reinforcement learning. This involves configuring the RL agent to interface directly with the simulated student; the RL agent provides formal state descriptions and tutorial strategy decisions to the simulated student, and the simulated student returns successor states that emulate state transitions one would expect to observe from actual human students. In addition, the simulated student generates estimates of student learning outcomes, which serve as reward values to drive reinforcement learning. In this manner, an RL-driven tutorial planner can sample different tutorial policies by repeatedly observing the effects of its pedagogical decisions on the simulated student's behavior and learning. As the RL agent samples different policies over many training episodes, it can prioritize more promising areas of the tutorial policy space and ignore areas that are unlikely to yield positive student learning outcomes. In contrast to training a tutorial planner with human student data, the RL agent has access to virtually unlimited training data. It is constrained only by the validity of the simulated student model, as well as the computational resources available for reinforcement learning (e.g., compute cycles, memory).

While training RL-driven tutorial planners with simulated students has several attractive characteristics, creating high-quality simulated students raises a broad range of issues inherent to user modeling. These include selecting the appropriate grain-size for the student model, leveraging an appropriate computational framework, managing the simulation model's complexity, modeling the desired facets of learner behavior, and maintaining the efficiency of the model so that RL can be applied on available computing hardware. In the next section, we survey related work on simulated students, including research on using simulated users for training RL-driven adaptive software systems.

GENERATING SYNTHETIC TRAINING DATA WITH SIMULATED STUDENTS

For several decades, the intelligent tutoring systems community has explored applications of simulated students for adaptive learning environments (VanLehn, Ohlsonn, & Nason, 1993; Beck, Woolf, & Beal, 2000). Recent years have seen growing interest in simulated students, as exemplified by a recurring series of workshops co-located with the International Conference on Artificial Intelligence in Education (AIED-13, AIED-15). An early paper by VanLehn, Ohlsonn, and Nason (1993) outlined three practical applications of simulated students. First, simulated students can be utilized to provide teachers with practice opportunities for refining their instruction. Second, simulated students can serve as co-learners for human students, providing opportunities for collaborative learning. Third, instructional designers can utilize simulated students to conduct formative evaluations of learning materials. In the intervening years, several additional applications have emerged. For example, the SimStudent project has demonstrated that simulated students can be used to author intelligent tutoring systems, using both authoring-by-tutoring and authoring-by-demonstration paradigms (Matsuda, Cohen, & Koedinger, 2014). Another application of simulated students is to generate synthetic data for inducing computational models of intelligent tutoring, which is the subject of our current work (Beck, Woolf, & Beal, 2000; Folsom-Kovarik, Sukthankar, & Schatz, 2013).

Approaches to creating simulated students vary along several dimensions. An especially important dimension is representational grain size. In his seminal work on the Soar cognitive architecture, Allen Newell (1990) described four bands of cognition—biological, cognitive, rational, and social—which organize different time scales for analyzing human action within a unified theory of cognition. Years later, Anderson (2002) revisited these bands, drawing connections between the fine-grained time scales of biology (e.g., milliseconds) and cognitive psychology (e.g., seconds) to coarser grained representations of time scale relevant to education (e.g., 100 hours). Similarly, simulated students operate at varying levels of temporal granularity. Fine-grained simulations have been devised to model human learning at the level of individual knowledge components (Matsuda, Cohen, & Koedinger, 2014). The SimStudent project leverages a production rule representation of procedural knowledge for problem solving in cognitive tutors, which is theoretically based on the ACT-R model of human cognition (Anderson et al., 2004). In contrast, simulated students have been created at the grain size of entire academic programs. The SimGrad project saw the creation of a prototype simulated doctoral program that accounted for courses taken, grades received, enrollment and graduation dates, and other high-level facets of academic performance (Lelei & McCalla, 2015). Most research on simulated students is situated in between these two extremes, focusing on generation of student responses to pedagogical actions at the level of coarse-grained knowledge concepts.

There are also a broad range of computational frameworks for encoding simulated students. Some simulated students are authored as expert systems with hand-crafted models of knowledge and problem solving (VanLehn, Jones, & Chi, 1992). Other simulated students are represented as simple mathematical functions, such as weighted sums (Frost & McCalla, 2015) that have a small number of hand-selected parameters. These models vary in the extent to which they are theoretically grounded, or in same cases, they are based upon the intuitions of the system's designers. Another family of simulation approaches utilize machine learning to induce models of student behavior from corpora of human student data (Beck, Woolf, & Beal, 2000). These models are often effective at capturing the overall statistical distribution of a population of student behaviors, but they risk producing individual episodes that are inconsistent or non-sensible (MacLellan, Matsuda, & Koedinger, 2013).

An important characteristic of student simulations is their degree of model complexity. Model complexity impacts a simulation's capacity to account for individual differences in learning rates, problem-solving strategies, and student traits. VanLehn, Ohlsonn, and Nason (1993) distinguish between (1) tabular simulations and (2) algorithmic simulations. Tabular simulations can be implemented with look-up tables that specify the full range of possible behaviors of the simulated student. In contrast, algorithmic simulations are defined by comparatively complex procedures for generating synthetic student behavior, including behaviors that have not been explicitly encoded by the system's designer. Tabular simulations are efficient to run, straight forward to inspect, and intuitive to author, but they have limited capacity to generalize to unseen situations. Algorithmic simulations vary widely in their efficiency, and examining their behavior requires additional effort, but they are well suited for generalizing to novel situations.

To date, a majority of research on simulated students has focused on cognitive aspects of student learning. This includes simulations that predict students' problem-solving behaviors, learning outcomes, and academic performance (Beck, Woolf, & Beal, 2000; Matsuda, Cohen, & Koedinger, 2014; Rosenberg-Kima & Pardos, 2015). There is comparatively little work investigating simulations of students' emotional, motivational, and metacognitive processes, despite their strong representation in research on user modeling. Sabourin et al. (2013) investigated simulations of students' affective dynamics to investigate whether off-task behavior is an effective emotion regulation strategy in a game-based learning environment for middle school science education. Frost and McCalla (2013) investigated the social effects of peer learners in a simulated environment that recommended personalized sequences of learning objects.

Despite recent interest in applications of RL to education, there are only a small number of examples of simulated students utilized to generate synthetic data for training RL-driven intelligent tutoring systems. The ADVISOR intelligent tutoring system used temporal-difference learning to induce teaching policies from corpora of synthetic student data (Beck, Woolf, & Beal, 2000). The synthetic data were generated by a Population Student Model (PSM), which was comprised of a logistic regression model induced with student data from a series of earlier classroom studies. The PSM generated predictions about how likely a student was to answer a problem correctly and how long the student would take to provide a response. ADVISOR was able to devise teaching policies that reduced human students' problem-solving times and generalized across distinct student populations for a grade school mathematics tutor. More recently, work by Folsom-Kovarik, Sukthankar, and Schatz (2013) devised a partially observable Markov decision process (POMDP) framework for intelligent tutoring, which was evaluated in a Call For Fire task for the U.S. Marine Corps. The framework introduced several techniques for improving the tractability of POMDP-based intelligent tutoring, including *state queues* and *observation chains.* Folsom-Kovarik et al. (2013) utilized simulated students to evaluate different POMDP representations for the underlying learner model.

Since there is a dearth of research on simulated students in RL-driven intelligent tutoring systems, related work on simulated users for intelligent interfaces is of considerable interest. Notably, simulated users serve a key role in RL-driven spoken dialogue systems, and they are particularly important in dialogue management (Schatzmann, Weilhammer, & Young, 2006). Early work on simulated users for spoken dialogue systems was typically linguistically motivated and relied heavily on knowledge-based formalisms (Schatzmann et al., 2006). Similar to educational applications, hand-authoring parameter values in dialogue management is challenging, because human dialogue behavior is often ill understood, population-dependent aspects of dialogue are difficult to estimate, and human authoring is rife with bias. Contemporary research on simulated users for dialogue systems is largely statistical, and it focuses on modeling users with machine learning techniques. Statistical models such as n-grams, Bayesian networks, and hidden Markov models have shown considerable promise for devising effective user simulations for training RL-driven dialogue managers (Schatzmann et al., 2006).

TRAINING TUTORIAL PLANNERS WITH SIMULATED STUDENTS IN GIFT

We are investigating data-driven tutorial planning in GIFT for two COIN training environments: (1) the

UrbanSim Primer hypermedia-based learning environment, and (2) the UrbanSim simulation-based learning environment. These two learning environments are complementary. Typically, learners will complete several units of the UrbanSim Primer to familiarize themselves with foundational COIN concepts, and afterward they will complete a series of training scenarios in the UrbanSim simulation. Both of these learning environments have been integrated with GIFT in order to serve as a general testbed for creating and evaluating RL-driven tutorial planners for COIN training.

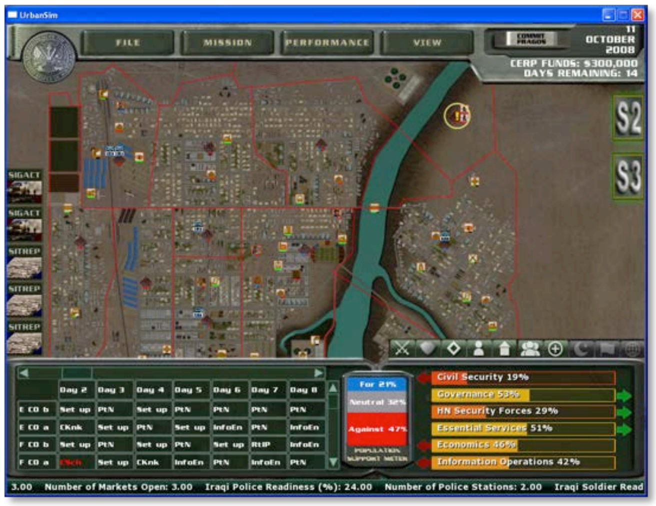
The UrbanSim Primer is a hypermedia-based learning environment that provides direct instruction on complex counterinsurgency and stability operations. Developed by the USC Institute for Creative Technologies, the UrbanSim Primer presents hyperlinked video, audio, text, and diagrams on a range of doctrinal concepts of counterinsurgency, including the importance of population support, strategies such as Clear-Hold-Build, resources and processes for intelligence gathering, and issues in successful execution of COIN operations. The Primer also provides preliminary instruction on the usage of the UrbanSim simulation.

UrbanSim is an open-ended simulation-based virtual training environment for counterinsurgency and stability operations (Fig. 2). In UrbanSim, learners act as a battalion commander whose mission is to maximize civilian support for the host nation government. Training experiences using UrbanSim resemble computer gameplay interactions with turn-based strategy games. On each turn, the learner assigns actions for 11 Battalion resources, such as “E Company, A platoon patrols the Malmoud Quarter” or “G Company, B platoon recruits policemen in the Northern Area.” Trainees’ actions, and consequences to their actions, are simulated using an underlying social-cultural behavior engine that determines how the host city’s inhabitants respond to different situations.

Generalized Instructional Strategies for COIN Training

We selected four types of instructional strategies for modeling and delivery by RL-driven tutorial planners for COIN training. The instructional strategies enable a common encoding of pedagogical actions across both the UrbanSim and UrbanSim Primer learning environments. The instructional strategies include (1) single-topic coaching, (2) multi-concept review, (3) feedback on unproductive learning behaviors, and (4) no feedback. These instructional strategies are delivered to learners using GIFT’s Tutor User Interface (TUI).

Single-topic coaching consists of a text-based feedback message about a specific dimension of learner performance in either the *security* or *meetings with host-nation leaders* performance areas of COIN training. These feedback messages can be delivered at the end of any turn in UrbanSim, or alternatively, at the end of an UrbanSim Primer unit, which is presented in the form of a PowerPoint show. Multi-concept reviews are similar to single-topic coaching, except that they address multiple dimensions of COIN performance simultaneously. Multi-concept reviews interleave summaries of effective COIN operational practice and excerpts from relevant U.S. Army field manuals, such as the *Commander’s Handbook for Strategic Communication and Communication Strategy*. Feedback on unproductive learning behaviors focuses on addressing egregious or inefficient actions performed by



	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

3.00 Number of Markets Open: 3.00 Iraqi Police Readiness (%): 24.00 Number of Police Stations: 2.00 Iraqi Soldier Read

Fig. 2. UrbanSim simulation-based training

learners that have little or no relevance to the learning task within UrbanSim or the UrbanSim Primer.

For each of these instructional strategies, we select among three possible variants for implementing the strategy. The variants are based upon the ICAP framework (Chi, 2009), which distinguishes between (1) interactive, (2) constructive, (3) active, and (4) passive forms of instructional activities. Our project does not focus on "interactive" forms of instructional methods, which typically refer to tutorial dialogues, so we have devised instructional strategies consistent with constructive, active, and passive forms of each technique. In other words, each of the three instructional techniques in this project—single-topic coaching, multi-concept reviews, and unproductive learning behaviors—has passive, active, or constructive variants.

The passive form of an instructional technique consists solely of a text-based message that participants read prior to continuing with their training. Passive instructional strategies do not require a particular response from the learner beyond clicking a button at the conclusion of the feedback message, but they are efficient and enable learners to promptly return to hypermedia or simulation-based training. The active form of an instructional technique expands upon the passive strategy by prompting learners to highlight key parts of feedback, or review the message, to identify its most important elements. After the learner completes her highlight, she is presented with an expert highlight of the same instructional message in order to facilitate critical evaluation of her own active learning performance. The constructive form of an instructional technique expands further by prompting learners to briefly summarize, in their own words, the most important parts of the feedback or review message. After the learner finishes writing her summary, she is presented with an expert summary in order to facilitate her own evaluation of her learning performance.

Designing Generalized Simulated Students for COIN Training Environments

In order to create data-driven tutorial planners for COIN training, we are currently devising simulated students to emulate behavior patterns and learning outcomes of human students interacting with the training system. We have a small dataset from an initial pilot test (N=23) conducted with ROTC cadets using UrbanSim and the UrbanSim Primer, which is informing efforts to manually author coarse-grained student simulations for each of the two learning environments. Both sets of student simulations are encoded in tabular format as probability mass functions; their format is closely related to the state-transition and reward models specified in MDP models of tutorial planning. This format was chosen because it is sufficiently granular to provide synthetic data for reinforcement learning, and it is highly efficient for generating large volumes of synthetic data. Devising a more fine-grained simulation is beyond the scope of the project, because granular cognitive-task analyses have not been conducted for UrbanSim and UrbanSim Primer.

Each simulated student is designed as a *bipartite model*. First, it consists of a joint probability distribution characterizing stochastic transitions between tutorial planner states. Second, it includes a joint probability distribution characterizing student learning outcomes from terminal states. The probability values in these models are informed by aggregated observations of state transitions and learning outcomes from the pilot test data. However, data sparsity issues require manual estimation of missing probability values. These model parameters will be validated and refined as additional data is collected from human students during the project. For UrbanSim, the temporal grain size for a simulated student corresponds to a single turn of the training simulation. For UrbanSim Primer, the grain size corresponds to a single lesson, which is typically a few minutes in duration. The student simulations focus on modeling cognitive and behavioral facets of student learning.

We model high-level decisions about pedagogical strategies in terms of three binary AESs: a single-topic coaching AES, a multi-concept review AES, and an unproductive behavior feedback AES. Each of these AESs is modeled by a distinct MDP, and similarly, it is associated with its own bipartite simulated student model. In addition, each of the aforementioned AESs is associated with a lower-level AES that encodes decisions about ICAP-inspired implementation strategies. In other words, if the tutorial planner chooses to deliver single-topic coaching, the planner's control flow transitions to a follow up MDP that selects among passive, active, and constructive variants of the coaching intervention. If the tutorial planner chooses to deliver a multi-concept review, the planner's control flow moves to a different MDP for passive, active, constructive decisions.

The state features for simulated students draw upon several sources: (1) student mastery of relevant knowledge concepts, (2) relevant task states, (3) learner attributes, and (4) pedagogical history. Because the MDP models interface primarily with GIFT's Pedagogical Module, their state representations are restricted to domain-independent features. The same is true of state representations for the simulated students. The selection of specific state features for the simulated students is ongoing, but the choice of features will seek to balance between model complexity and expressiveness. The actions that each simulated student responds to are the pedagogical actions associated with each AES. Rewards will correspond to discretized COIN content learning gains for UrbanSim Primer and simulation training performance for UrbanSim.

In practice, each simulated student will be instantiated with several different configurations of parameters, allowing the simulated student model to reflect a population of student learners, rather than behaviors of a single student. We intend to investigate the effects of alternate parameterizations of simulated students on the learning rates and policies yielded for the RL-driven tutorial planner. In addition, we intend to qualitatively analyze the resulting tutorial policies in light of current theory on instructional design and learning science.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

Data-driven approaches to tutorial planning, such as reinforcement learning, show significant promise for devising effective models of instructional strategies for complex domains and learning environments. We are investigating RL-driven tutorial planning in the domain of COIN training, with a focus on the UrbanSim Primer and UrbanSim learning environments. We have presented a brief survey of the research literature on simulated students, which suggests that model granularity, choice of computational framework, model complexity, modeled learning behavior, and efficiency are key factors that distinguish different types of simulated students. Leveraging findings from this review, we are currently devising generalized simulated students for COIN training, which will be used to generate synthetic data for training RL-driven tutorial planners in GIFT.

There are several promising directions for future work. Conducting studies to validate simulated students by comparing synthetic data with actual human behaviors will be an important step. Devising tools, workflows, and examples for incorporating tutorial planning policies induced from simulated students in GIFT is planned. In addition, providing tools for non-experts to work with simulated students, including creating, configuring, sharing, and refining simulated student models, holds potential to significantly expand research on simulated students and advance GIFT's objectives of realizing low cost, automatically generated instruction across a broad range of training domains.

ACKNOWLEDGMENTS

The research described herein has been sponsored by the U.S. Army Research Laboratory under cooperative agreement W911NF-15-2-0030. The authors would like to thank the faculty and students of the NC State University Army ROTC program for their assistance this research. The statements and opinions expressed in this article do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

REFERENCES

Anderson, J. R. (2002). Spanning seven orders of magnitude: A challenge for cognitive modeling. *Cognitive Science*, 26, 85-112.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of mind. *Psychological Review, 111*(4), 1036-1060.

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

Chi, M.T.H. (2009). Active-Constructive-Interactive: A conceptual framework of differentiating learning activities. *Topics in Cognitive Science*, *1*(1), 73-105.

Chi, M., VanLehn, K, Litman, D. & Jordan, P. (2011). An evaluation of pedagogical tutorial tactics for a natural language tutoring system: A reinforcement learning approach. *International Journal of Artificial Intelligence in Education*, 21(1-2), 83–113.

Folsom-Kovarik, J.T., Sukthankar, G. & Schatz, S. (2013). Tractable POMDP representations for intelligent tutoring systems. *ACM Transactions on Intelligent Systems and Technology*, 4(2), 1–22.

Frost, S. & McCalla, G. (2015). An Approach to Developing Instructional Planners for Dynamic Open-Ended Learning Environments. *Proceedings of the Workshops at the 17th International Conference on Artificial Intelligence in Education (AIED-15)*, Madrid, Spain, pp. 1–10.

Lelei, D. E. & McCalla, G. (2015). Exploring the Issues in Simulating a Semi-Structured Learning Environment: The SimGrad Doctoral Program Design. *Proceedings of the Workshops at the 17th International Conference on Artificial Intelligence in Education (AIED-15)*, Madrid, Spain, pp. 11–20.

MacLellan, C., Matsuda, N., & Koedinger, K. (2013). Toward a Reflective SimStudent: Using Experience to Avoid Generalization Errors. *Proceedings of the AIED 2013 Simulated Learners Workshop*. Memphis, TN, pp. 51–60, Springer.

Matsuda, N., Cohen, W.W., & Koedinger, K.R. (2014). Teaching the Teacher: Tutoring SimStudent Leads to More Effective Cognitive Tutor Authoring. *International Journal of Artificial Intelligence in Education*, 25(1), 1–34.

Mislevy, R. J., Oranje, A., Bauer, M. I., von Davier, A., Hao, J., Corrigan, S., Hoffman, E., DiCerbo, K., & John, M. (2014). *Pyschometric Considerations in Game-Based Assessments*. Retrieved from https://www.ets.org/research/policy_research_reports/publications/white_paper/2014/jrrx

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University.

Rosenberg-Kima, R. & Pardos, Z., 2015. Is this Model for Real? Simulating Data to Reveal the Proximity of a Model to Reality. *Proceedings of the Workshops at the 17th International Conference on Artificial Intelligence in Education (AIED-15)*, Madrid, Spain, pp. 78–87.

Rowe, J. & Lester, J. (2015). Improving Student Problem Solving in Narrative- Centered Learning Environments: A Modular Reinforcement Learning Framework. *Proceedings of the 17th International Conference on Artificial Intelligence in Education*, Madrid, Spain, pp. 419-428.

Sabourin, J., Rowe, J., Mott, B., & Lester, J. (2013). Considering Alternate Futures to Classify Off-Task Behavior as Emotion Self-Regulation: A Supervised Learning Approach. *Journal of Educational Data Mining*, 5(1), 9–38.

Schatzmann, J., Weilhammer, K., Stuttle, M., & Young, S. (2006). A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The Knowledge Engineering Review*, *21*(2), 97–126.

Shute, V. J., Ventura, M., & Kim, Y. J. (2013). Assessment and Learning of Qualitative Physics in Newton's Playground. *The Journal of Educational Research*, *106*(6), 423–430.

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

VanLehn, K., Jones, R. M., & Chi, M. T. (1992). A model of the self-explanation effect. *The Journal of the Learning Sciences*, 2(1), 1–59.

VanLehn, K., Ohlsson, S. & Nason, R. (1993). Applications of Simulated Students: An Exploration. *Journal of Artificial Intelligence in Education*, 5(2), 1–42.

Williams, J. J., Kim, J., Rafferty, A., Maldonado, S., Gajos, K., Lasecki, W., & Heffernan, N. (2016). AXIS: Generating Explanations at Scale with Learnersourcing and Machine Learning. *Proceedings of the 3rd Annual ACM Conference on Learning at Scale*, Edinburgh, SCT, pp. 379-388.

Woolf, B. P. (2008). *Building Intelligent Interactive Tutors: Student-Sentered Strategies for Revolutionizing e-Learning*. Morgan Kaufmann.

ABOUT THE AUTHORS

Dr. Jonathan Rowe *is a Research Scientist in the Center for Educational Informatics at North Carolina State University. He received the Ph.D. and M.S. degrees in Computer Science from North Carolina State University, and the B.S. degree in Computer Science from Lafayette College. His research is in the areas of artificial intelligence and human-computer interaction for advanced learning technologies, with an emphasis on game-based learning environments, intelligent tutoring systems, user modeling, educational data mining, and computational models of interactive narrative.*

Dr. Bob Pokorny *is Director of the Education and Training Technology Division at Intelligent Automation, Inc. He earned his Ph.D. in Experimental Psychology at University of Oregon in 1985, and completed a postdoctoral appointment at University of Texas at Austin in Artificial Intelligence. Bob's first position after completing graduate school was at the Air Force Research Laboratory, where he developed methodologies to efficiently create intelligent tutoring systems for a wide variety of Air Force jobs. At Intelligent Automation, Bob has led many cognitive science projects, including adaptive visualization training for equipment maintainers, and an expert system approach for scoring trainee performance in complex simulations.*

Dr. Benjamin Goldberg *is a member of the Learning in Intelligent Tutoring Environments (LITE) Lab at the U.S. Army Research Laboratory's (ARL) Human Research and Engineering Directorate (HRED), Simulation and Training Technology Center (STTC) in Orlando, FL. He has been conducting research in modeling and simulation for the past five years with a focus on adaptive learning and how to leverage artificial intelligence tools and methods for adaptive computer-based instruction. Currently, he is the LITE Lab's lead scientist on instructional strategy research within adaptive training environments. Dr. Goldberg is a Ph.D. graduate from the University of Central Florida in the program of Modeling & Simulation.*

Dr. Bradford Mott *is a Senior Research Scientist in the Center for Educational Informatics at North Carolina State University. Prior to joining North Carolina State University, he served as Technical Director at Emergent Game Technologies where he created cross-platform middleware solutions for video game development, including solutions for the PlayStation 3, Wii, and Xbox 360. Dr. Mott received his Ph.D. in Computer Science from North Carolina State University in 2006, where his research focused on intelligent game-based learning environments. His current research interests include computer games, computational models of interactive narrative, and intelligent game-based learning environments.*

Dr. James Lester *is Distinguished Professor of Computer Science at North Carolina State University, where he is Director of the Center for Educational Informatics. His research centers on transforming education with technology-rich learning environments. With a focus on adaptive learning technologies, his research spans intelligent tutoring systems, game-based learning environments, affective computing, and tutorial dialogue. The adaptive learning environments he and his colleagues develop have been used by thousands of students in K-12 classrooms. He received his Ph.D. in Computer Science from the University of Texas at Austin in 1994. He is a Fellow of the Association for the Advancement of Artificial Intelligence (AAAI).*