Educational Data Mining Using GIFT Cloud

Alan Carlin[1], Evan Oster[1], Chris Nucci[1], Diane Kramer[1], Keith Brawner[2]
Aptima, Inc.[1], Army Research Laboratory[2]

INTRODUCTION

Intelligent Tutoring Systems (ITS), like human instructors, make frequent decisions about what to present to the student. These decisions include what courses or content to present next, as well as what type of After Action Review (AAR) to present to the student after each course. Ideally, the AAR would be Adaptive (AAAR). In this work, we analyze the decisions of course content and presentation. We construct a student model which models the skills necessary, the effectiveness of each course at training each skill, and the relationship between in-scenario measures and student skill-level. If the student model is accurate and represented mathematically, then decision-theory can be used by the ITS to select courses and course content.

There are two ways to develop the mathematical model of skills, effectiveness, and transition. A first way is that a Subject Matter Expert (SME) or an instructor can carefully build it, using an interface that translates SME intuition to model parameters. In this work, however, we explore tools to facilitate a second option, that of building the model automatically from a corpus of student data. We report on progress towards an enhanced version of the Newtonian Talk tutor (Zhao et al., 2015), on GIFT Cloud, using these mathematical modeling methods. Enhancements include the ability to select an AAR adaptively, the ability to display that AAR, and the ability to sequence courses in a customized order. Each of these requires an ability to learn information about the training domain. To this end, we report on a data collection study which will produce the information necessary to build the enhanced tutor.

DATA COLLECTION METHOD

First, a mathematical mapping of skills and effectiveness was created from data. In order to facilitate the modeling process, data was collected on 44 subjects using the GIFT-Powered NewtonianTalk tutor (Zhao et al., 2015). Prior to the experiment, participants were asked to sign an informed consent form, complete a brief survey, and take a pretest. The survey gathered demographic information such as age, education level, gender, and average physics grade. The pretest consisted of 8 questions based on figures and gathered data on participant’s physics knowledge. An example is shown at the left of Figure 1. Once the survey and pretest were completed, the data collection began.

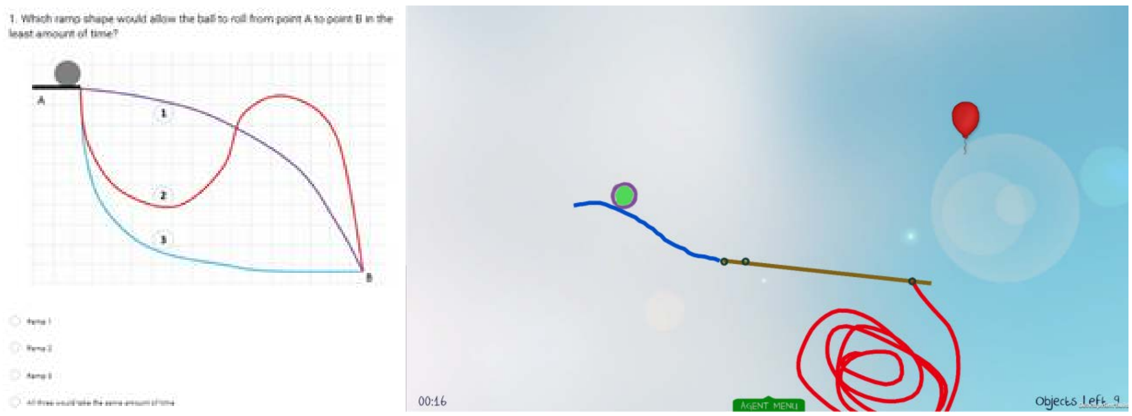

Figure 1: (Left) Pre-test question within Newtonian Talk tutor. (Right) Courses within Newtonian Talk.

During data collection, participants completed a series of physics puzzles in Newton's Playground (an example is shown at the right of Figure 1). This is a game that presents learners with puzzles to solve by strategically creating physics-based objects in a 2D virtual space in order to manipulate a ball, which pops a balloon. Figure 1 shows the ball in green, the red balloon in the upper right, and a series of student-drawn objects in both blue and red.

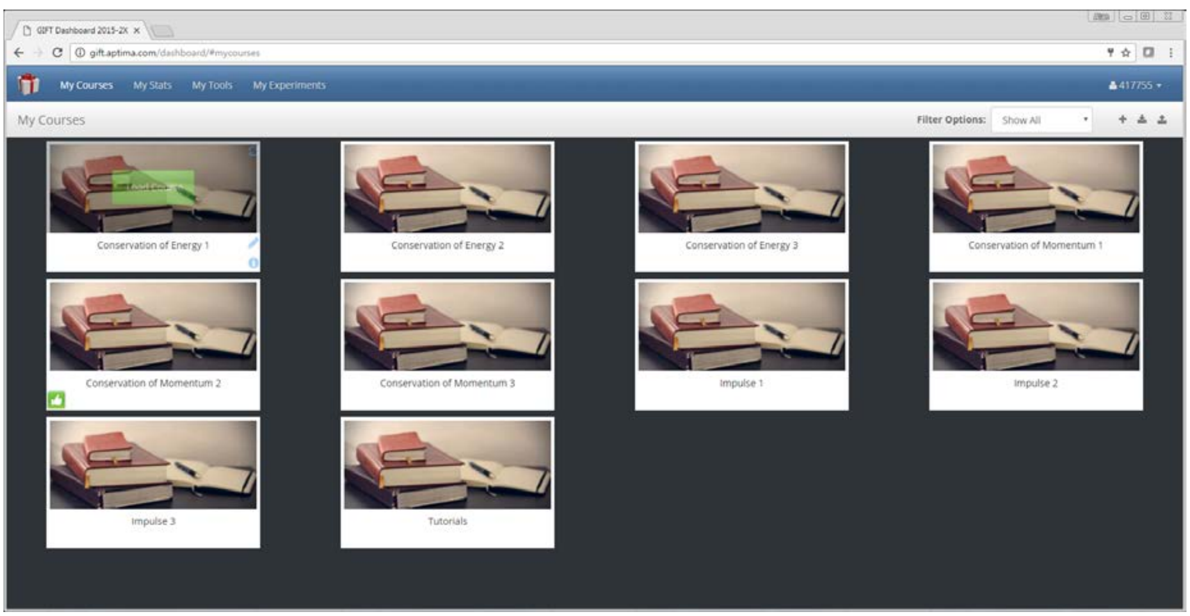

Figure 2: Selection screen that includes a Tutorial and 9 Newtonian Talk Courses.

Participants were randomly assigned a unique user ID which designated a unique path through the series of 9 such puzzles. As part of the reported effort, we modified the GIFT software so that one particular course was recommended at random (shown by the green thumb on the leftmost course in the second row of Figure 2). Subjects were instructed to simply always select the recommended course.

At the end of the session, participants completed an 8 question posttest similar in nature to the pretest to gather data as a comparison point of physics knowledge to the pretest. This was followed by a short debriefing where they learned more about the purpose of this data collection. The data collection lasted about

45 minutes to 1 hour, but the exact amount of time depended upon learning pace. Learners were able to take breaks at any time during the session.

IMPACT ON GIFT FRAMEWORK

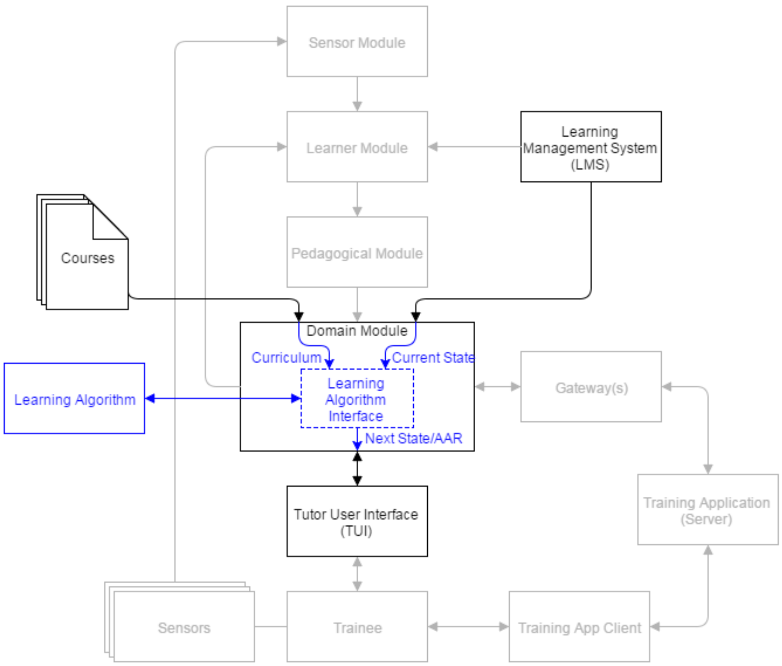

Figure 3: Modules affected within the GIFT framework.

Support for the data collection entailed several modifications to the GIFT framework. Figure 3 highlights the modules affected by the data collection. The workflow through the framework is as follows:

1. The student takes a pre-test to assess physics knowledge.
2. An adaptive learning algorithm (e.g., MDP) determines the next learner course.
3. An AAR screen shows the results of the learning algorithm.
4. The learning algorithm also makes a recommendation the students' next course.
5. The student takes the next course.
6. The process repeats (go to step 2 above) 9 times.
7. After the 9th scenario, the learning loop ceases and the student takes a post-test.

The contributions of this framework lie in Steps 2,3, and 4. To support the data collection and a follow-up effort, we made modifications to GIFT software and Newtonian Talk so that course content and AAR (an AAAR) was personalized to the individual. These modifications included:

- The Learning Management System (**LMS)** was modified to store a custom construct called student state, described further in the next section. Related to the LMS modification, messages were added throughout the system so that state could be transmitted between modules.
- The **Domain Module** was enhanced to include classes that represent an adaptive policy, as well as the logic for utilizing the adaptive policy and using the policy to return AAR information. The Domain Module was also used to get/set user state in the LMS.

- **Tutor UI**: The tutor UI was tweaked to display custom AAR information.

These modifications all represented improvements on previous work towards customizing GIFT (Hruska et al., 2011). In the next section, we discuss the State data structure used by both the LMS and Domain Module. The modifications to the Tutor UI will be discussed later in this document.

DATA MINING MODULE

To model the data, we used a Partially Observable Markov Decision Process (POMDP; Smallwood & Sondik, 1973). The POMDP model contains various parameters which must be identified for the specific domain: State, Actions, an Observation function (measures), Transitions, and Reward. Since Reward is usually assigned by a human and reflects the individual instructor's priorities, we will not discuss it further in this document. The other parameters are data mined by the AAR system, and are described below.

State

Definition 1 (State): State is defined as a k-tuple of numbers, with k representing the number of skills in the training domain. For the preliminary Newtonian Talk study, we automatically extracted the course measures from the LMS, and we considered each measure available to the system to measure its own skill. In future studies, this requirement will be relaxed so that each measure does not necessarily need to measure one skill. An example of a state is $< 3,5 >$, referring to a skill level of "3" on the first skill and "5" on the second. The Newtonian Talk domain, since it had 10 measures, had 10 skills. Symbolically, we represent state with the symbol θ.

Definition 1a (State Probability): We refer to the probability of being in a state with the notation Pr(). E.g., Pr(<1,1>) represents probability of being in state <1,1>. We may also denote a given point in time when the student was in that state. That is, $Pr(< 1,1 >^{\{t=0\}})$ refers to the probability that a student was in state <1,1> at time zero.

Actions

The set of Actions was identified as the set of 9 courses in the Newtonian Talk tutor. For each course we associated an id, and we created variables to represent the applicability and difficulty of the course to each component of state. The variable d_i^k was used to refer to the difficulty of course i with respect to skill k, and the weight w_i^k was used to refer to the applicability weight of course i with respect to skill k.

Observation Function

To fit the model, we used Item Response Theory (IRT; Lord, 1980) to fit the observation parameter. In Newtonian Talk, each course was either passed or failed, and we expressed the probability that a student would pass a course as $Pr(correct)$. IRT models performance on items using logistic regression.

Equation 1 (IRT):

$$p(correct) = \frac{1}{1+e^{(d_i-\theta)}}$$

Thus, in IRT, the probability of correct completion of a course depends on the course difficulty and the student state. If the course difficulty exceeds the student state, the student is unlikely to complete the course correctly. Conversely, if the student state exceeds the course difficulty, then the probability of completion is high.

The AAAR framework extends the notation by vectorizing it to account for many skills. Let k identify a skill. We modify Equation 1 so that the overall capability of the student to perform on the item, is the sum of the capabilities on the individual skills. This yields:

Equation 2: $$p(correct) = \frac{1}{1+e^{\sum_k w_i^k (d_i^k - \theta_j^k)}}$$

Where d_k^i, and w_i^k have been introduced above, and where $\theta_{\boldsymbol{j}}^k$ represents the skill level of student j on skill k. Equation 2 differs from Equation 1 in that probability of completion is now a linear weighted combination of difficulties and skill levels. When we want to discuss all skills of a student, we will use a vector, so we would represent all skills of student j with a bar to represent a vector, as in $\overline{\theta_J}$, or an alternative is boldface, $\boldsymbol{\theta_j}$.

Example 1: Suppose a student is at level 3 for skill 1, and level 5 for skill 2. We summarize this by saying $\boldsymbol{\theta} =< 3{,}5 >$. Suppose item 22 is at difficulty level 6 for skill 1, and 2 for skill 2. That is, $d_{22} =< 6{,}2 >$. Plugging back into Equation 2, the student is modeled as 50% likely to get the item correct.

Transition Function

We would like to model students that improve as they train, following on the literature of deliberate practice (Ericsson et al., 1993). In the above model, there is only one student variable for each skill θ_j^k, instead we would like to break this out into several variables $\theta_j^{k,t}$ representing the skill level at skill k, by student j, at time t. Our model does not use a specific time like 53.45151 seconds, but rather discretizes into time steps. In Newtonian Talk, t represents the number of courses completed by the student thus far. That is we model student skill after 0 courses, after 1 course, after 2 courses, etc.

We can then model a *transition function*, which we denote as *T*, and represents the probability of student improvement. The transition function takes the form:

Expression 3: $$T\left(\theta_j^{k,t+1} \middle| \theta_j^{k,t}, \alpha\right)$$

This represents the probability of that student j achieving a skill level on skill k the next step (that is, at time $t+1$), given that student's skill level at the current time step (represented by time t), and the training action (i.e., the course) α.

For the current AAR model, we assume transitions are independent between skills. This eventually will not need to be the case, and if transitions were not independent we would use:

Expression 4 $$T\left(\theta_j^{t+1} \middle| \theta_j^{t}, \alpha\right)$$

We propose two methods to assign this probability. The simpler method is to directly interpret transition probability as an artifact of item difficulty levels and student states through a rule: the closer a course's difficulty to a student's skill level, the more likely the course is to train the student. This corresponds to Vygotsky's Zone of Proximal Development (Vygotsky, 1978). The second is to solve for these probabilities directly based on machine learning the value-assignments for all of the other variables and counting the number of transitions. In this study, we explore this second option.

Goal of Data Mining Study

The purpose of the data mining study was to learn values for all difficulty variables d, all weights w, all transition probabilities $T(\theta_j^{t+1}|\theta_j^t, \alpha)$ for the Newtonian Talk tutor. Variable values were learned by a Gibbs Sampling algorithm (Geman & Geman, 1984). Values for subject knowledge states θ_j^{t+1}were also learned for the subjects of the data collection study. Knowledge of the domain variable values will allow for the construction of adaptive training algorithms in future studies. The adaptive training algorithms will optimize course selection based on these learned variable assignments.

PRELIMINARY RESULTS

The data collection was completed in February 2017. In this paper, we report on a preliminary analysis.

Data Mining Result

We used domain information from Newton's Playground as well as Gibbs sampling to sample values for the variables discussed in the above section. To facilitate, we defined measures and skills synonymously (i.e., each measure observes a single unique skill). If a measure/skill was present in a course, the course was assigned a "1" for presence of that skill. A summary of activities and puzzles is shown in Figure 4. Figure 4 shows which courses are available, the subjects that they intend to teach, and the activities required to complete them.

Course	Name	draw pin	draw spring	draw anything	draw pinned	draw freeform	draw ramp
Playground 1, Puzzle 1	Tutorial 1						
Playground 1, Puzzle 2	Tutorial 2						
Playground 1, Puzzle 3	Tutorial 3						
Playground 1, Puzzle 4	Tutorial 4						
Playground 1, Puzzle 5	Tutorial 5						
Playground 2, Puzzle 1	Impulse 1	X	X				
Playground 2, Puzzle 2	Impulse 2	X	X				
Playground 2, Puzzle 3	Impulse 3	X	X				
Playground 3, Puzzle 1	Momentum 1			X	X		
Playground 3, Puzzle 2	Momentum 2			X	X		
Playground 3, Puzzle 3	Momentum 3			X	X		
Playground 4, Puzzle 1	Energy 1					X	X
Playground 4, Puzzle 2	Energy 2					X	X
Playground 4, Puzzle 3	Energy 3					X	X

Figure 4: Summary of activities and puzzles. For each puzzle (rows), the related skills/measures are denoted.

After all skills were assigned as present/not present, the course was normalized so that the sum of its applicability variables was 1. Based on the data, skill level was assessed for each of the subjects in the data collection on a 1-10 scale. Below, the assessment is shown for the first several courses of a typical student. Each row of the table represents assessed student state on the various skills, after the r-th course, where r is the row number in the leftmost column.

Course #	General	Draw Anything	Draw Freeform	Draw Pin	Draw Pinned	Draw Ramp	Draw Spring	User Ramp	User Spring
0	2	1	1	2	2	0	0	0	1
1	3	1	1	3	3	0	0	0	1
2	3	2	2	3	3	1	1	0	2
4	3	3	2	5	4	1	1	0	2
5	7	3	2	5	5	2	2	1	2
6	7	4	3	6	5	2	2	2	3
7	8	5	4	7	5	2	2	3	3

Figure 5: Assessed state for one of the subjects of the data collection study. Subject skill level progressed after each course (row).

Overall course difficulties were estimated (on a 1-10 scale) based on sampled fit to the item response equation. Below shows the last sample taken. In future analysis, the average of the samples will be retained. As an example, in Figure 6 the course "p2p1.course.xml" was assigned a difficulty level of 7 as a result of the Gibbs sampling learning process.

Course ID	Sampled Diffiulty
p2p1.course.xml	7
p2p2.course.xml	6
p2p3.course.xml	6
p3p1.course.xml	4
p3p2.course.xml	5
p3p3.course.xml	4
p4p1.course.xml	1
p4p2.course.xml	5
p4p3.course.xml	5
tutorials.course.xml	1

Figure 6: Assessed difficulty of various Newtonian Talk courses on a 1-10 scale.

Transition probabilities were found as well, although these values were too numerous to list in a table. The transition data was 4-dimensional: each course, skill, and pre-course state, each post-course state was assigned a probability. The table below shows the probability of attaining various skill levels for the p2p1.course.xml, when the student was at a skill level of 0 before the scenario.

0	1	2	3	4	5	6	7	8	9
71.1%	28.3%	.6%	0	0	0	0	0	0	0

Figure 7: Transition probability for one course, from a starting skill level of zero. Each column represents probability of attaining a new skill level as a result of the course. Reference Figure 4 for the mappings between skills and actions.

Simulated Student Result

Using the learned model, it is possible to simulate students using desired instructional strategies. For example, Figure 8 compares simulated student progress using an adaptive training strategy that intelligently selects NewtonianTalk puzzles, versus a strategy that selects random puzzles. To generate this figure, 10,000 students were modeled by the POMDP produced by the data mining procedure discussed in this paper. Each student transitioned randomly to a new state after each scenario, according to a distribution governed by the POMDP transition function. The adaptive strategy selected the best available puzzle for that given student, whereas the random strategy selected a random puzzle. Figure 8 represents average skill level attained, across all skills (see columns of Figure 4), across a 10000 students.

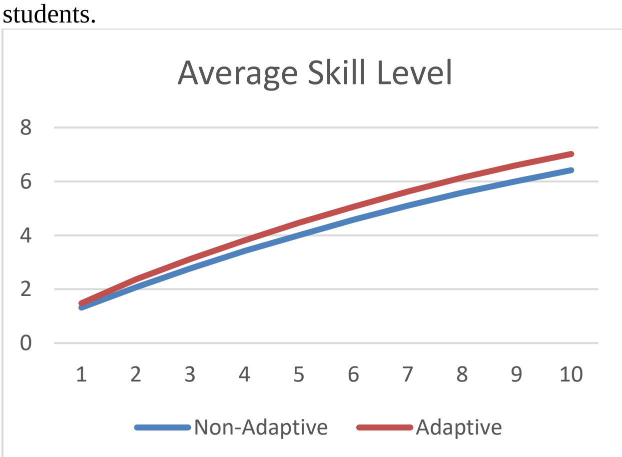

Figure 8: Simulated student skill level using an adaptive training strategy versus using a random one. The x-axis represents number of simulated puzzles, y-axis represents student skill level on a 1-10 scale, as produced by the model.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

In this paper we have reported on a data collection study. The study modified the Newtonian Talk branch of GIFT to sequence courses, to store results in the LMS, and to interpret information information about the courses using data mining techniques. A preliminary analysis of the data is described in this paper. With the data collection complete, there are several future directions which will all take place over the next year.

1. Refine and enhance the analysis of the variables described in this study.

2. Use the resulting variable values to parameterize an adaptive training algorithm, and use this algorithm to sequence subjects in Newtonian Talk for a future study, thus proving the efficacy of the GIFT framework on adaptive training.

3. Use student assessments to present an After Action Review. The mockups below show examples of what this After Action Review will look like for future experiments.

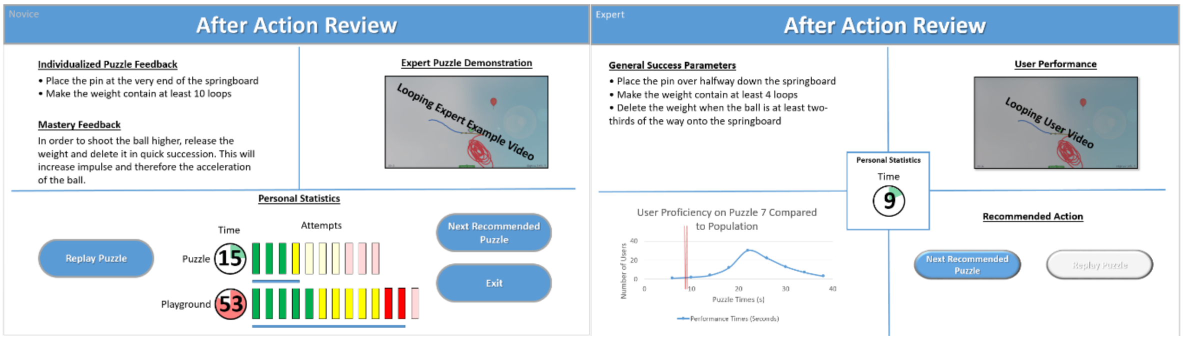

Figure 9: Different AARs are pictured for different users, based on experiences and data mined policy.

REFERENCES

Ericsson, K. A., Krampe, R. T., & Tesch-Römer, C. (1993). The role of deliberate practice in the acquisition of expert performance. Psychological review, 100(3), 363.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. Pattern Analysis and Machine Intelligence, IEEE Transactions on, (6), 721-741.

Hruska, M., Dewey, M., Paonessa, G., Paonessa, M., Poeppelman, T., Nucci, C., & Ayers, J. (2013). Software Design Document (SDD) for Soldier Performance Planner Application. Report to US Army RDECOM Army Research Laboratory – STTC. Problem Solutions.

Lord, F.M. (1980). Applications of item response theory to practical testing problems. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Smallwood, R. D. & Sondik, E. J. (1973). The optimal control of partially observable Markov processes over a finite horizon. *Operations Research, 21*, 1071-1088.

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

Vygotsky, L. (1978). Interaction between learning and development. Readings on the development of children, 23(3), 34-41.

Zhao, W., M. Ventura, et al. (2015). GIFT-Powered NewtonianTalk. Generalized Intelli-gent Framework for Tutoring (GIFT) Users Symposium (GIFTSym3), Orlando, FL.

ABOUT THE AUTHORS

Alan Carlin, PhD is a Senior Research Engineer at Aptima, Inc. His interests focus on problems of artificial intelligence and machine learning. These include problems of decision making under uncertainty, communication between members of a team, and meta-reasoning for decision-makers. His publications include works on Decentralized Partially Observable Markov Decision Processes, game theory, and distributed meta-reasoning in uncertain environments. While at Aptima, he has developed intelligent training systems, NextGen pilot alerting systems, and data mining systems. Dr. Carlin received a Ph.D. in Computer Science from the University of Massachusetts, an M.S. in Computer Science from Tufts University, and a dual B.A. in Computer Science and Psychology from Cornell University. As part of his M.S., he also completed the MIT Lincoln Scholar's Program, sponsored by the Massachusetts Institute of Technology.

Evan Oster is an Associate Scientist at Aptima, Inc. with experience in courseware development, game-based training, and augmented reality-based instruction. His current work centers around using cognitive science and instructional design to increase training effectiveness in live and synthetic military environments. He has experience managing interdisciplinary teams responsible for the development and integration of various training products. Past work has involved designing game and simulation-based training using immersive virtual environments for the Navy. Mr. Oster has also created several augmented reality games with a university games lab while providing software development evaluation. Evan Oster holds an M.A. in Instructional Design and Technology and a B.S. in Human Development from Virginia Polytechnic Institute and State University as well as an M.S. in Curriculum and Instruction from Radford University.

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

Diane Kramer is a Principal Software Engineer and Certified Scrum Master at Aptima, Inc. with over 25 years of experience designing and developing software applications using various programming languages and platforms. Additionally she has academic experience teaching Computer Science at both the college and high school levels. Her current work at Aptima involves leading small teams of engineers, and developing applications for adaptive training, working with scientists who encode algorithms such as Partially Observable Markov Decision Process (POMDP) and Best Fit Optimization (BFO) models. One example is a Small Business Innovative Research (SBIR) Phase II project involving developing a training platform for Full Motion Video Imagery Analysts. Ms. Kramer re-ceived a M.S. in Computer Science from Worcester Polytechnic Institute, and a B.A. in Computer Science from the University of Massachusetts, Boston. She is a member of the Association of Computing Machinery, and the national Computer Science Teachers Association.

Chris Nucci is a Senior Software Engineer at Aptima, Inc. with engineering experience in a variety of fields including live training, gunnery, cyber security, secure coding. His current work focuses on the ongoing development of SPOTLITE™, Aptima's tablet-based application for performance measurement, data collection, debrief, and analysis. Other work has included development on Aptima's PM Engine™ and Army's GIFT framework. His previous experience at Lockheed Martin includes development of a Storage Area Network (SAN) configuration service for the National Cyber Range (NCR), development of gunnery and gallery target and mover controller software for the Saudi Arabian National Guard (SANG) Range Modernization program, exercise planning and AAR tools for U.S. Army Program Executive Office for Simulation, Training, and Instrumentation (PEO STRI) Common Training Instrumentation Architecture (CTIA), Live Training Transformation (LT2), and Combat Training Center (CTC), as well as experience in secure coding, databases, and web application development. Mr. Nucci received a M.S. and B.S. in Computer Science from the Florida Institute of Technology.