The Thirty-First International Florida
Artificial Intelligence Research Society Conference (FLAIRS-31)

Data Mining for Adaptive Instruction

Alan S. Carlin,[1] Chris Nucci,[1] Evan Oster,[1] Diane Kramer,[1] Keith Brawner[2]

[1]Aptima, Inc. {acarlin, eoster, cnucci, dskramer}@aptima.com
[2]Army Research Laboratory, keith.w.brawner.civ@mail.mil

Abstract

Adaptive tutoring is an interesting optimization problem due to being a noisy domain with intermittent feedback. Noise in the domain hinders efforts to understand past data, and intermittent feedback makes it hard to optimize current decisions. In this paper we describe an end-to-end adaptive tutoring system that addresses these challenges; it closes the loop between data mining, generation of an adaptive tutoring policy, and After Action Review (AAR). The AAR tutoring system described in this paper uses a physics tutor, called Newton's Playground, to develop the technology. This paper reports on a pilot study of 9 participant-experienced scenarios where the data was automatically analyzed, then an adaptive training policy was created based on the analysis. Furthermore, an AAR was constructed and delivered. The end result of this process is an enhanced adaptive version of Newton's Playground, run on a preliminary set of new subjects.

Introduction

Human-to-human tutoring has the potential for between one and two standard deviations of improvement over traditional learning (Bloom 1984; VanLehn 2011). There are many parts to intelligent tutoring such as examination of affective state, assessment of learner ability and knowledge, and selection of additional content. Among the problems of Intelligent Tutoring Systems (ITS) is selection of content, referred to as the "outer loop" and the selection of feedback, typically referred to as the "inner loop" (Van Lehn 2016).

Typically, solutions for "outer loop" content selection involve the selection of the next part of the content as it maps to the learning objectives of the learners. However, often in a learning system the learner does *not* learn the material upon the first experience, making remediation necessary. The decision of whether to remediate, and the selection of remedial content, is a commonly used portion of the outer loop selection. The problem is difficult, as each learner is somewhat unique and presents relatively sparse data to the system.

Testing an individual technology, such as an "outer loop" content selection algorithm, requires a significant substrate of technology (e.g. the domain of instruction, modeling technique for the learner, instructional intervention timing logic, etc.). The Generalized Intelligent Framework for Tutoring (GIFT) is an empirically-based, service-oriented framework of tools, methods and standards to make it easier to author computer-based tutoring systems (CBTS), manage instruction, and assess the effect of CBTS, components and methodologies. GIFT is a "shell tutor" – a tutoring system used to make tutoring systems (Brawner et al. 2017). GIFT depends on a number of interchangeable "Modules" which encapsulate knowledge of the Domain, Learner, Pedagogy, Simulation, and Sensors, respectively. In this manner, the system is equipped to tutor on many different subjects with little additional system configuration. As an intentional byproduct of this structure, software developed for training one domain (e.g. mathematics) can be reused to train a domain which is completely different (e.g. combat lifesaver policies). As a result, the development of improved "outer loop" remedial content selection technologies for one domain of instruction can be readily applied to many other training domains. Similarly, an After Action Review (AAR) technology can be developed with minimal overlap with other ITS functions.

The intelligent AAR selection problem addressed in this paper is relatively unique in the area of machine learning. Because of the nature of GIFT, the first of the problem aspects is that the developed solution must have an instructional policy decoupled from instructional material. Further, the solution must handle sparse data gracefully – each student is relatively unique and generates only a small amount of data on which to make decisions.

The remainder of this paper describes the related work with adaptive after action reviews, the Physics Playground training environment, the process of creation of AI models, initial data collection, and concludes with an evaluation of the model and approach.

Related Work

The approach in this paper combines information about the effectiveness of instructional content with information about the learner. The former is grounded in Ericcson's theory of deliberate practice (Ericsson 1993), by targeting training scenarios based on learner skill states.

One approach that can be useful for quantifying the learner state is Bayesian Knowledge Tracing (BKT; Corbett and Anderson 1995). The BKT model has 4 parameters, representing the probability that a student has a skill (*init*), the probability that the student acquires the skill after content (*transit*), the probability that the student makes a mistake despite having the skill (*slip*), and the probability of guessing correctly (*guess*) when not having the skill. When performance is observed, the model can be represented as a Hidden Markov Model (HMM).

Another approach is Item Response Theory (IRT). In IRT, the probability of a learner getting an item correct is derived by quantifying the learner skill level for some skill labeled i (s_i) and the item difficulty level (d), subtracting the former from the latter, and sending the result through a logistic function, e.g. the probability of correctness is:

$$\frac{1}{1 + e^{(d-s_i)}}$$

The above represents the 2-parameter model, but there are other multi-parameter IRT models representing multiple skills (Wirth & Edwards 2007).

The model in this paper expands upon BKT by using it in a prescriptive context. That is, instead of probability of improvement and measurement for a single skill (an HMM transition probability), we represent multiple *actions*, where each action can pertain to many skills. It also combines BKT with IRT through the measurement function. These expansions model learning as a POMDP (Partially Observable Markov Decision Process) rather than just an HMM.

There are several previous works in the literature which also use a POMDP representation (Sondik 1971). Brunskill and Russell (2012) use a POMDP on a smaller algebra domain, but they represent skill acquisition as a binary and do not combine with IRT. Folsom-Kavarik uses POMDP to model observation chains and priorities queues based upon help-seeking activities (Folsom-Kovarik 2012). Rowe and Lester (2015) model learning as several MDP's, examining each skill separately from the others in environments where progress can be fully observed.

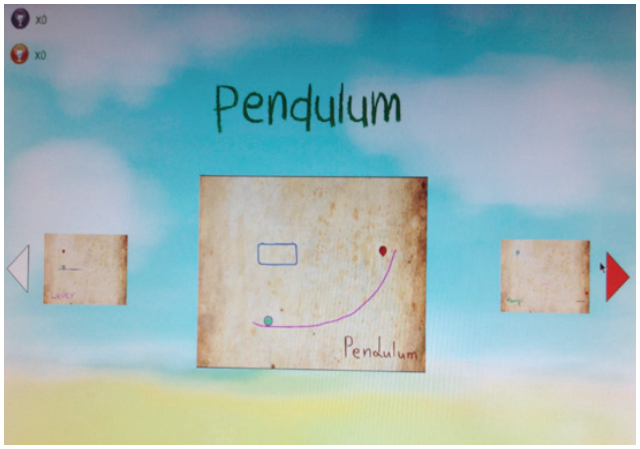


Figure 1: Sample of the Newton's Playground environment.

Approach

Newton's Playground is a "serious game" – a game designed to teach as well as entertain (Zhao et al. 2015; Kim et al. 2016). During the experience of the game, the learners/players draw different types of physics items, such as a lever, weight, or structural beam. Each of these items is then animated according to the basic laws of physics.

The goal of each level is to get a red ball from one point on the screen to another with a system of drawn objects. An example puzzle may intend to teach the concept of spring physics by having the learner draw a "diving board" type structure consisting of drawn objects (beam, two anchor points, a spring).

This work uses a prototype version of Newton's Playground built for GIFT, where puzzles are instrumented with measures including the completion time for each puzzle. 9 puzzles based on three physics concepts (Impulse, Conservation of Momentum, Conservation of Energy) were constructed using this technology.

Figure 2 shows the approach taken. One contribution of this work is the implementation of a POMDP Policy for selection of training within GIFT. This policy selects feedback (more on this in subsequent sections) which is delivered to the Learner. The data is recorded in GIFT's Learning Management System (designated "Data Repository" in the figure). A second contribution of this effort is an Educational Data Mining (EDM) tool to understand this data.

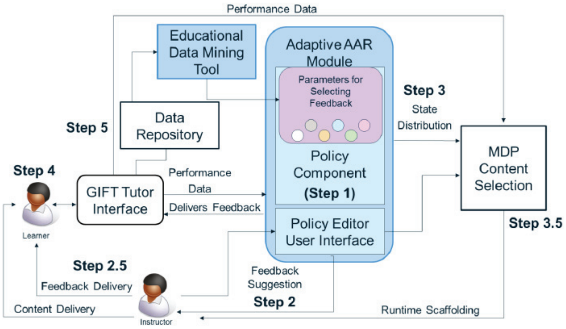


Figure 1: Modeling process for policy creation.

The EDM tool models the learning process as a POMDP tuple $< \boldsymbol{S}, \boldsymbol{A}, \boldsymbol{Pr}, \boldsymbol{\Omega}, \boldsymbol{O}, \boldsymbol{R}, \boldsymbol{b_0} >$.

- $\boldsymbol{S}$ is the set of states. Each $\boldsymbol{s} \in \boldsymbol{S}$ is described as $\boldsymbol{S_1} \ldots \boldsymbol{x}\ \boldsymbol{S_n}$ where each $\boldsymbol{S_i}$ represents the learner skill level for one skill (we chose a scale from 0-9), and $\boldsymbol{n}$ is the number of skills.
- $\boldsymbol{A}$ is the set of Actions. In this domain, we model the tutor's selection of each puzzle in Newton's Playground as an action.
- $\mathbf{Pr}\ (\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s'})$ is the state transition model given the learner state and the selected action. In a simple model, as in this work, the transition probability can be described as the product of transitionally independent individual skills; so that $\mathbf{Pr_i}(\boldsymbol{s'_i} | \ \boldsymbol{s_i}, \boldsymbol{a})$, where $\boldsymbol{i}$ identifies a skill, $\boldsymbol{s_i}$ represents current learner state and $\boldsymbol{s'_i}$ represents successor learner state. For more complicated domains, state is still decomposed into individual skills, but the relationship between skills after each action is defined using a more complete Dynamic Bayesian Network (DBN) containing skill and action nodes.
- $\boldsymbol{\Omega}$ is the set of possible observations. For Newton's playground puzzles, this is measurement of whether a student passes or fails a lesson given information acquired from GIFT Domain Knowledge Files (DKF).
- $\boldsymbol{O}$ the observation function that assigns a probability $\boldsymbol{O}(\boldsymbol{o} \in \boldsymbol{\Omega} | \boldsymbol{a}, \boldsymbol{s'})$ as Equation (1), with $\boldsymbol{s'}$ being the learner state
- $\boldsymbol{R}(\boldsymbol{s}, \boldsymbol{a})$ is a reward model. We model reward as the sum of the skill levels across skills.
- $\boldsymbol{\gamma}$ is a discount factor applied to reward that reduces reward on each subsequent step.
- $\boldsymbol{b_0}$, an initial distribution over $\boldsymbol{S}$ before the student begins.

The EDM tool inputs historical student performance data, and outputs parameters to fill in the above model. POMDP's are typically solved by applying a Bellman update. Defining $\mathbf{b}$ as a probability distribution over states, the value of that distribution is summarized as:

$$\boldsymbol{V}(\boldsymbol{b}) = \max_{\boldsymbol{a} \in \boldsymbol{A}} [\boldsymbol{R}(\boldsymbol{b}, \boldsymbol{a}) + \boldsymbol{\gamma} \sum_{\boldsymbol{o} \in \boldsymbol{\Omega}} \boldsymbol{O}(\boldsymbol{o} | \boldsymbol{b}, \boldsymbol{a}) \boldsymbol{V}(\boldsymbol{b'})]$$

That is, the value of a distribution of states b is equal to the highest-value action, where the value of that action is the expected direct reward ($\boldsymbol{R}(\boldsymbol{b}, \boldsymbol{a})$; overloading the notation to apply an expected value) plus the discounted expected reward of the subsequent actions taken on the subsequent belief state (denoted $\boldsymbol{b'}$ and representing the new belief state after action a and observation o).

A property of the formulation in the EDM tool is that the observation function corresponds to Item Response Theory, that is:

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

Where each i represents a skill within the state, each $\boldsymbol{d_i}$ represents the difficulty of the item, each $\boldsymbol{a_i}$ represents the applicability of the item to the skill. Thus, the task of machine-learning the POMDP observation function is reduced to the task of machine-learning item difficulties.

On the AAR, following the literature we used principles of Focus, Style, Quantity, and Objective (Li 2010; Shute 2008; Billings 2012). On Focus, AAR systems should offer feedback near the level of the task step. On Style, feedback should be corrective. On Quantity, feedback should be limited to failures with greatest impact. On Objective, AAR should represent deep knowledge and associate it with task steps or cues.

Data Collection

The parameters for the model above were found by using an EDM tool operating on the results of a data collection study.

The data collection involved presenting 42 participants with 10 puzzles from Newton's Playground. The puzzle set contained a tutorial and 3 puzzles on Conservation of Energy, 3 puzzles on Conservation of Momentum, and 3 puzzles on Impulse. Contributions of this effort involved customized software to extract measures from each participant, in each puzzle, and storing this information in a Learning Management System (LMS).

There were two types of parameters that were learned from the recorded results. The first type was parameters that were identified through meta-data in Newton's Playground. These included the states (which corresponded to the names of the measures recorded in the LMS), the actions (9 actions, 1 for each puzzle), and the observations (students were recorded for *Pass/Fail* of the puzzles, with *Pass* decomposed into *AboveExpectation* and *AtExpectation*). The second type included variables that were approximated through Gibbs sampling of the data. This included the transition probabilities and the observation probabili-

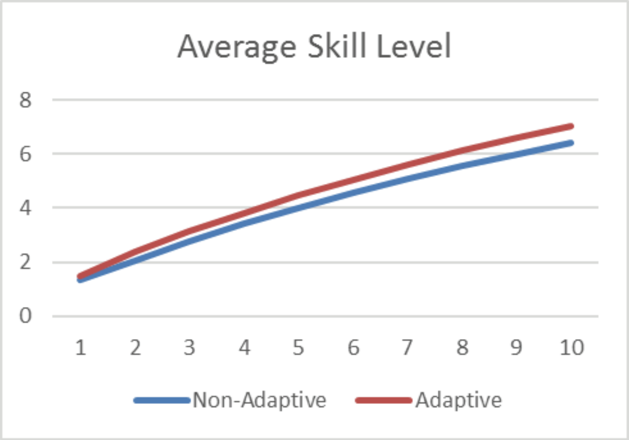


Figure 2: Simulated students for adaptive and non-adaptive puzzle selection

ties. The observation function was found by finding a best-fit of the item difficulty parameter to the results, given the set of data and the sampled values for the other variables in the POMDP model.

Simulation Results

After the model parameters were determined, a POMDP policy was generated that mapped each state to an action. We simulated 10000 students to validate the policy. The simulation included 10 steps, before the first step, student ability was sampled from a start distribution. Each simulated student iteratively took a puzzle (selected from a Tutorial or the 9 available puzzles), then an observation was received, and then the policy would select the next puzzle. Figure 3 shows an aggregate comparison (averaged over the 10000 students) of this policy to a Non-adaptive policy which selected random puzzles.

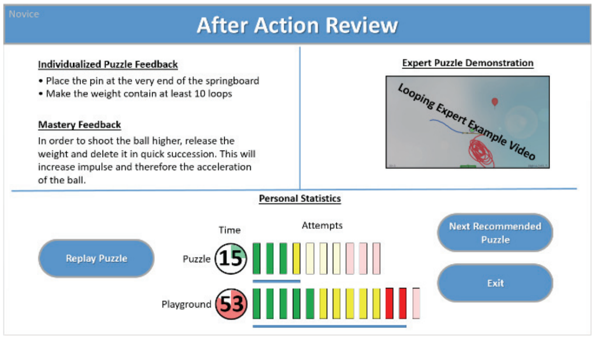
Subsequent Study

The parameterized model was also used to generate technology to enhance GIFT so as to improve learning outcomes. To measure improved outcomes, we selected 3 puzzles out of the 9 in Newton's Playground (one from each physics topic) and designated those as the "test" puzzles. So as to allow discretion for the adaptive strategy to select preferred puzzles, we generated 4-step adaptive strategies to select 4 puzzles.

Two enhancements were made for the experimental group, to adapt training. The first was an adaptive policy based on optimizing POMDP reward.

To create this, we set and discovered POMDP parameters as follows:

- State was a 3-tuple where each element corresponded to student skill level on Impulse, Conservation of Momentum, or Conservation of Energy Principles. For example, the state of <novice, novice, expert> would correspond to a novice with respect to solving the Impulse puzzles, novice on the Conservation of Momentum puzzles, and expert with respect to solving the Conservation of Energy principles puzzles.
- Each of the six available puzzles were designated as an action.
- The transition function was discovered through Gibbs sampling the probability distribution based on student skill levels at each time step. Each variable representing student states, the transition function, and the difficulty level with respect to IRT for the observation function was sampled in turn. A Dirichlet hyperprior was used to enforce two intuitions:
    - Skills tend to appreciate over time.
    - All other things being equal, small leaps in ability are more likely than large leaps.
- Observations consisted of *Above Expectation*/*At Expectation*/*Below Expectation* for each puzzle, based on time taken to solve the puzzle. This was reported by Newton's Playground as instrumented in the associated Domain Knowledge Files (DKFs) as a threshold over time taken to solve the puzzle.
- The observation function was discovered through Gibbs sampling by assuming knowledge of student skill levels at each step, and solving for item difficulty using the observation function structure described previously.
- Reward was set as the sum of the elements in the State tuple.



Individualized Puzzle Feedback

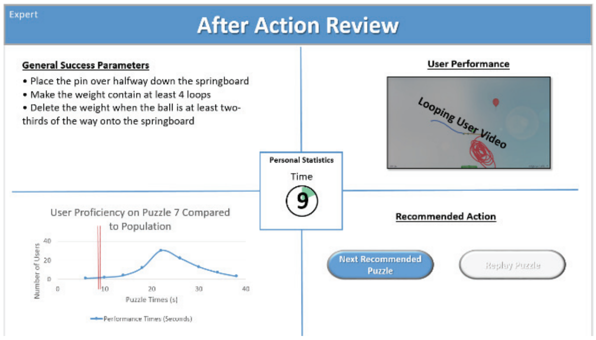
- Place the pin at the very end of the springboard
- Make the weight contain at least 10 loops

Mastery Feedback

In order to shoot the ball higher, release the weight and delete it in quick succession. This will increase impulse and therefore the acceleration of the ball.

Figure 3: Remedial AAR.

The second modification was the existence of an Adaptive AAR module, where a post-puzzle screen was shown after each of the 4 training puzzles. The Adaptive AAR is shown in Figure 4, with the AAR shown dependent on whether the student has obtained the skill yet. The construction of the AAR screens was guided by the literature on AAR construction, including to focus on the task-step (Van Lehn 2011), as well as required differences between high-achieving and low-achieving learners (Shute 2008).



- Place the pin over halfway down the springboard
- Make the weight contain at least 4 loops
- Delete the weight when the ball is at least two-thirds of the way onto the springboard

Figure 4: Advanced AAR.

For students that have not yet obtained the skill, the upper-left of Figure 4 contains advice on how to solve the puzzle, the upper-right shows a video of expert performance, and the bottom shows personal statistics and an ability to replay the puzzle or go to the next puzzle selected by the POMDP.

For those who succeeded on the puzzle (Figure 5), the upper-left of the Advanced AAR shows more general principles, the upper-right shows the user's own performance so they can refine their technique, the lower-left shows the user's performance as compared to the general population, and the lower-right prominently allows the user to go to the next puzzle while allowing the user to replay the puzzle if desired. In general, the goal of the Remedial AAR is to allow the user to learn how to pass, while the goal of the Advanced AAR is to refine performance.

Table 1 shows the first rows of the resulting policy after data mining and solving. Policies are state machines (note: "state machine" refers to the concept of executable software, not to be confused with "state" as used in the rest of this paper within a POMDP model). Each row references a particular node/state, and is grouped with the rows containing the same state. At the node, a puzzle is recommended. "Mo 1" stands for the first momentum puzzle. The "LB" and "UB" columns represent lower and upper-bound conditions under which the rule in the row is applied, with a "0" representing Below Expectation measured performance, a "1" representing "At Expectation" and a "2" representing Above Expectation. The "Next" column represents the next node/state to go to, and the AAR text references which text is next. Thus, the first two lines of the table say that the puzzle to be given is the tutorial no matter what, and the next node to go to is node 2. The lines that reference node 2 say that if the student performs Below expectation ("0") the Remedial feedback is given and the software goes to Node 3 (where the next puzzle is "Momentum 3"), if the student performs "At Expectation" ("1") the student still receives Remedial Feedback but goes to Node 4, whereas if the student is measured "Above Expectation" the student receives the Advanced feedback while going to Node 4.

node	puzzle	LB	UB	Next	AAR Type
1	Tutorial	0	1	2	
1	Tutorial	1	2	2	
2	Mo 1	0	0	3	Remed
2	Mo 1	1	1	4	Remed
2	Mo 1	2	2	4	Adv
3	Mo 3	0	0	10	Remed
3	Mo 3	1	1	10	Remed
3	Mo 3	2	2	10	Adv

Table 1: Adaptive AAR policy executed on GIFT.

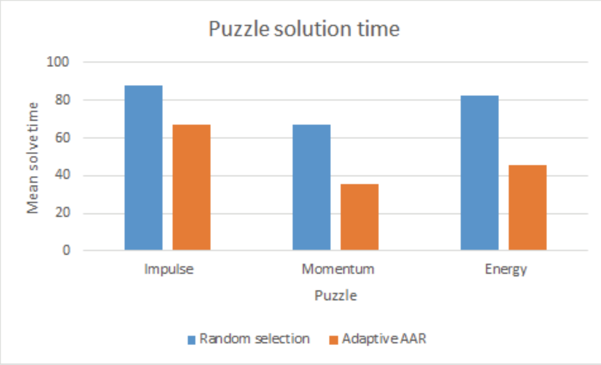


Figure 5: Mean solution time for Adaptive AAR versus random for three test puzzles.

Results and Evaluation

After the initial data collection, a new study was run using the same pool of 9 Newton's Playground puzzles. Data was collected on 8 participants. Data from the previous collection was used as a control condition. To provide adequate differentiation between the puzzles selected between the adaptive and control conditions, three puzzles were selected as test puzzles ("Impulse 3", "Momentum 2", and "Energy 3", in that order), and the remaining six puzzles comprised the pool of training puzzles for adaptation. Of these six, four were selected for training, in an adaptive sequence determined by the policy. Each participant in the adaptive condition encountered four puzzles, selected online by the adaptive policy, and then was tested on the three test puzzles that were withheld from the training pool. The random condition in Figure 6 shows the data from the previous collection. For apples-to-apples comparison, result times from the data collection were not used if the puzzle occurred at an earlier point in the sequence. For example, the Impulse test puzzle in Figure 6 was 5th in the sequence, so we only used results from the random condition where this same puzzle was encountered as 5th or later in the random sequence. As a result, 31, 19, and 17 participant results were used for Impulse, Momentum, and Energy puzzles respectively in the random condition.

Results are reported in Figure 6. To compute mean puzzle times, we assigned a value of 120 seconds to solutions that took longer than this time (puzzles that were not solved in two minutes were likely to take far longer). Differences (effect size) in median solve times between groups were 36, 39, and 45 seconds respectively, greater than the effect for mean solve time, indicating that the selection of a 120 second cutoff was not responsible for the effect. Standard deviations were 33, 35, and 39 respectively for the random selection condition puzzles, and 40, 30, and 39 respectively for the adaptive AAR condition puzzles. Results tended to be bimodal, participants who solved the puzzle usually completed it in far less than 120 seconds, while those who did not were assigned the 120 second maximum time.

Conclusion

Contributions of this work include: (1) An educational data mining tool for GIFT. (2) Generation and implementation of a model-based adaptive training policy for GIFT. (3) Implementation of After Action Review technology for GIFT.

The process detailed in this paper of modeling student progress in a manner independent from the instruction, testing this model with simulated learners, applying this model in real-time to learners, and observing initial effectiveness portrays a manner in which other systems can seek to adapt to learner material. The approach detailed in this paper offers unique benefits to both the GIFT architecture and GIFT end users for improving learning through adaptive, individualized feedback to the learner. First, the POMDP technique of monitoring and improving state corresponds to the theory of deliberate practice. Second, the model dynamically re-parameterizes while leveraging prior data to inform actions – the system improves over time with relatively few data points. Finally, utilizing GIFT, the solution is applicable to a wide variety of training domains.

References

Anderson, J. R.; Corbett, A. T.; Koedinger, K. R.; and Pelletier, R. 1995. Cognitive tutors: Lessons learned. *The journal of the learning sciences*, 4(2):167-207.

Biber, D., Nekrasova, T., & Horn, B. 2011. The Effectiveness of Feedback for L1-English and L2-Writing Development: A Meta-Analysis. *ETS Research Report Series.* No. 1.

Bloom, B. S. 1984. The 2-Sigma Problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher* 13(6):4-16.

Brawner, K.; Sinatra, A. M.; and Sottilare, R. 2017. Motivation and research in architectural intelligent tutoring. *International Journal of Simulation and Process Modelling 12*(3-4):300-312.

Brunskill, E.; Russell, S. 2010. RAPID: A reachable anytime planner for imprecisely-sensed domains. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*. Redmond, WA: AUAI Press.

Corbett, A.T.; and Anderson, J.R. 1995. Student modeling in the ACT programming tutor. *Cognitively diagnostic assessment*, 19-41.

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

Folsom-Kovarik, J. 2012. Leveraging Help Requests In Pomdp Intelligent Tutors. Ph.D. diss., Department of Computer Science, University of Central Florida, Orlando, FL.

Kim, Y. J.; Almond, R. G.; and Shute, V. J. 2016. Applying evidence-centered design for the development of game-based assessments in physics playground. *International Journal of Testing* 16(2):142-163.

Li, S. 2010. The effectiveness of corrective feedback in SLA: A meta-analysis. *Language Learning* 60(2):309-65.

Rowe, J. P.; and Lester, J. C. 2015. Improving student problem solving in narrative-centered learning environments: A modular reinforcement learning framework. In *International Conference on Artificial Intelligence in Education*, 419-428. Springer, Cham.

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

Sondik, E. 1971. The optimal control of partially observable Markov decision processes. Ph.D. thesis, Stanford University.

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

VanLehn, K. 2016. Regulative loops, step loops and task loops. *International Journal of Artificial Intelligence in Education 26*(1): 107-112.

Wirth, R. J.; and Edwards, M. C. 2007. Item factor analysis: current approaches and future directions. *Psychological methods* 12(1):58.

Zhao, W.; Ventura, M.; Nye, B. D.; and Hu, X. 2015. GIFT-powered NewtonianTalk. In *Generalized Intelligent Framework for Tutoring (GIFT) Users Symposium (GIFTSym3)*, 3.