Plan Ahead: Pricing ITS Learner Models

Jeremiah T. Folsom-Kovarik
Sae Schatz
Denise Nicholson
University of Central Florida
Institute for Simulation and Training
3100 Technology Parkway
Orlando, FL 32826
jfolsomk@ist.ucf.edu, sschatz@ist.ucf.edu, dnichols@ist.ucf.edu

Keywords:
intelligent tutoring systems, learner models, development costs

ABSTRACT: *Intelligent tutoring systems (ITSs) are highly adapted to individual learners, and therefore their learner models are central to their operation and account for a large fraction of their development costs. Different learner model architectures may have different development costs, but those costs are not widely reported in the literature. This paper presents individual reports from an anonymous questionnaire sent to ITS professionals in September 2009. The respondents estimated the development costs of recent ITSs and their associated learner models. The resulting data aligns with and amplifies published accounts, as well as contributing new cost information about model types that have not previously appeared in the literature.*

1. Introduction

In an intelligent tutoring system (ITS), personalized treatment makes teaching and training more effective. ITSs adapt their interactions to individual learners by estimating users' traits, states, or misconceptions in a learner model. Since adaptation and personalization play defining roles in ITSs, the learner model is key to every new system. Practitioners will benefit from an open discussion of what to expect when developing different types of models.

Following Snow and Swanson (1992), this paper divides personalization in an ITS into macroadaptation and microadaptation. *Macroadaptation* describes changes the ITS makes prior to a learning episode based on pre-task measures or historical data, which can include problem selection or ordering. *Microadaptation* describes changes during a learning episode based on ongoing performance or behavioral assessment, which can include giving the learner custom hints and feedback.

Several competing model architectures support ITS adaptation, and published accounts reviewed in this paper suggest that different model types might have different impacts on cost. To the extent that model types support macroadaptation and microadaptation, they can all be appropriate choices for an ITS. One factor that could help practitioners choose a learner model is its development cost. Controlling the cost of a model can make more resources available for other development tasks or help maximize the project's return on investment.

This paper compares anecdotes about the cost to develop different learner model architectures, as one important consideration among many in designing a new ITS. In the rest of this section, the model types being considered are introduced and information about their development in the published literature is reviewed. The remainder of this paper describes a questionnaire of the ITS community that solicited additional anecdotes focused on development costs. Section 2 explains the questionnaire method, section 3 describes the results, and section 4 provides some interpretation of these results.

1.1 Model architectures

This section introduces six learner model architectures common in the ITS field. These architectures form the separate categories described in this paper.

An *overlay* metaphor describes the earliest and simplest learner models, such as those in Scholar (Carbonell, 1970), PLATO West (Burton & Brown, 1976), and Wusor II (Carr, 1977). An overlay model is conceptually like a checklist of all the knowledge and skills an ITS must impart. The ITS records learners' competencies as a subset, or overlay, of the ideal checklist. It gives a novice no checkmarks and a perfect expert a checkmark for each item on the list. Successful or unsuccessful performance in the tutor grows or shrinks the overlay. *Differential* models are a subset of overlay models that apply the "checklist" approach but do not require learners to master all expert knowledge to satisfy learning requirements.

Although overlay models can encode novices' lack of expert skills or knowledge, novices do not simply lack knowledge. Often, they also possess incorrect knowledge that an ITS should specifically identify and correct. *Buggy* or *perturbation* learner models include information about possible misconceptions or *bugs*. Model builders can either generate possible misconceptions automatically by systematically breaking rules in a cognitive theory, (e.g., Brown & VanLehn, 1980; Burton, 1982), or can let subject-matter experts list likely misconceptions (e.g., Johnson, 1990).

An extension of early buggy-model ITSs is the *cognitive tutor*. Like in buggy models, cognitive tutors model misconceptions as breaks in a cognitive model, but they also specify an algorithm, *model tracing*, for matching observed mistakes to the underlying misconceptions. The learner model in a cognitive tutor is a set of *production rules*, grounded in cognitive theory, that mirror the mental steps the learner makes while working—for example, selecting a theorem to apply in a geometry proof (Anderson, 1993). Model tracing tries different productions together to see which could have produced the learner's observed behavior. The granularity of the production rules supports detailed microadaptation but does not readily enable macroadaptation. To compensate for this, modern cognitive tutors typically also use a second learner model for macroadaptation, such as an overlay (Corbett & Bhatnagar, 1997) or Bayesian model (Baker, Corbett, & Aleven, 2008).

Model-tracing tutors revolve around a detailed cognitive model describing how learners work and learn. One way of building an ITS with similar performance, but with less cognitive science, is with *example tracing* (Koedinger, Aleven, Heffernan, McLaren, & Hockenberry, 2004). Instead of general cognitive rules that apply to any problem, example-tracing tutors let builders write example solutions for each problem. Specific errors can still trigger specific remediations, but only when examples of those errors are programmed ahead of time.

Another way to avoid reconstructing hidden mental events is to use a *constraint-based* model (Ohlsson, 1992). These models are collections of constraints, i.e., boundary conditions that describe incorrect problem states. Tutors based on constraints allow learners to interact freely with the system until something happens that requires correction. Uniquely among the learner models, constraint-based models assume that behaviors they do not recognize are correct—not wrong—and that learners are "innocent until proven guilty" (Mitrovic, Koedinger, & Martin, 2003, p. 320). Like production-rule models, constraint-based models can be paired with an overlay model to control macroadaptation, for example by inferring unmastered skills from constraint violations (Martin & Mitrovic, 2002).

Finally, *classifiers* can also play the role of a student model. A classifier as a learner model typically sorts individual learners into groups. These groupings can be similar to assessments from overlay or buggy models, but unlike typical overlay or buggy models, classifiers use more principled methods of interpreting observations as evidence, and potentially can update many model estimates with each assessment. Classifiers that have been used as learner models include *Bayesian networks* (e.g., Arroyo, Woolf, & Beal, 2006; Conati & Zhou, 2004; Luckin & du Boulay, 1999), *finite-state automata* (e.g., Stottler, Fu, Ramachandran, & Vinkavich, 2001), *decision trees* (e.g., Cha et al., 2006; McQuiggan, Mott, & Lester, 2008), *neural networks* (e.g., Castellano, Mastronardi, Di Giuseppe, & Dicensi, 2007), and ensemble methods (e.g., Hatzilygeroudis & Prentzas, 2004; Lee, 2007). Although there are many different kinds of classifiers, in at least some practical situations they are approximately equivalent in their performance (McQuiggan et al., 2008; Walonoski & Heffernan, 2006).

1.2 Published accounts

Although development cost is an important consideration for practitioners making an ITS operational, it is only irregularly reported in the academic literature. This section gathers reports that authors volunteered in published academic sources. The common metric for reporting ITS costs in these sources is the ratio of ITS development time in person-hours to user interaction time in hours per individual. Reporting costs in a ratio format makes figures more comparable across different ITSs that may undertake more or less complex tutoring tasks.

Cognitive tutors and model-tracing algorithms have been the subject of both significant research and also operationalization (Koedinger, Anderson, Hadley, & Mark, 1997). Initial publications on the first cognitive tutors reported cost ratios between 1000:1 and 100:1 to build an entire ITS (Anderson, 1993). As another example within this range, an algebra tutor had a 200:1 ratio for the whole system (Koedinger et al., 2004). Building cognitive tutors in the future may be easier because specialized authoring tools are in development. A preliminary study of a new authoring tool showed a 40% reduction in effort that could make future cognitive tutors more cost-effective (Aleven, McLaren, Sewall, & Koedinger, 2006).

Example tracing models were created as a response to the high development cost of using the model-tracing approach, and a preliminary study showed that cost ratios were only 23:1 for an entire example-tracing ITS (Koedinger et al., 2004). Furthermore, example tracing is more straightforward than model tracing for nonprogrammers, and novices could use it to build a whole ITS with a cost ratio of 40:1 (Razzaq et al., 2008).

Constraint-based tutors were also designed to require less development effort than cognitive tutors, because the tutor can still give meaningful results without a complete set of constraints or in domains for which it is difficult to write exhaustive production rules (Mitrovic et al., 2003). The first ITS based on constraints had a 220:1 cost ratio for building the learner model only (Mitrovic & Ohlsson, 1999). Since then, new authoring systems have let novices create a simple tutor or reimplement an existing ITS about as quickly as experts had previously (Martin, Mitrovic, & Suraweera, 2008; Mitrovic et al., 2006; Suraweera, Mitrovic, & Martin, 2007).

Constraints and production rules have also been directly compared on the cost of developing the same learner model. In one study, an expert in model-tracing built a cognitive tutor to teach the same domain as an existing constraint-based tutor. The two tutors were approximately equal in complexity and presumably in development cost (Mitrovic et al., 2003). In another study, a single team built new constraint-based and model-tracing tutors to teach the same task. They found that the constraint-based tutor took four times as long to implement because of extra effort to learn the more complex architecture. Excluding their learning time, the team found that model tracing took slightly more time to implement, but the two architectures nonetheless required approximately the same effort (Kodaganallur, Weitz, & Rosenthal, 2005).

While precise cost figures have not typically been published for overlay models, buggy models, or classifiers, some studies have explored these development experiences. For example, studies of buggy models suggest that generating a complete misconception list can be a long or even unending task because different misconceptions are prevalent in different populations. (Payne & Squibb, 1990; VanLehn, 1982). Theory also warns about potential high costs of Bayesian models. Initializing Bayesian networks can require precise expert estimates or large amounts of empirical data, although it is possible to start using the model with initial settings and refine it during use (Conati & Maclaren, 2005). The design effort grows quickly with complexity, so that a Bayesian network with just 40 inputs would be difficult to initialize, and its estimates would be highly suspect (Ott, Imoto, & Miyano, 2004).

The research community has produced limited reports on development time, including a comparison of the same team developing two equivalent model types and a comparison of experts in their respective architectures developing equivalent models. However, publication of development cost estimates remains sparse, with only a few estimates published for some model types and none at all for other widely used architectures. The rest of this paper helps to address these gaps in the published knowledge.

2. Method

2.1 Questionnaire

To increase knowledge of learner model development costs, an anonymous questionnaire was emailed to ITS community members in September of 2009. Because of space restrictions, only the parts of the questionnaire that produced data used in this paper are reproduced in the appendix. However, a full version of the questionnaire is available in (Folsom-Kovarik, Schatz, & Nicholson, in preparation).

The questions answered in this paper describe participants' experiences on the last ITS each person worked on that is ready or almost ready to interact with learners. This makes practitioners' memories more recent and also helps ensure the data presented reflect current modeling and authoring technology. Participants were asked to estimate the development effort in person-hours for the ITS as a whole and also for the learner model or models specifically. To calibrate the complexity of the ITS being described, participants were also asked the amount of time one learner would be expected to engage with the ITS. All questions were optional.

Participants were asked thirty additional questions relating to previous experiences with building specific model types. Because of low response rates and space limitations those questions are not discussed in this paper.

2.2 Participants

The questionnaire was emailed to all 63 attendees of the 2009 Army Research Institute Workshop on Adaptive Training Technologies and to an additional 88 authors of publications cited in a survey of the ITS field (Folsom-Kovarik et al., in preparation) who did not attend the workshop. Eleven participants responded anonymously. The responses give a varied anecdotal view of the development costs for different student models in the current state of the field.

Participants in the study came from diverse backgrounds. Of the eleven participants, five people were academics, three worked in industry, and two worked in government or military positions. Three people had worked on one or two ITSs, three had worked on three to five ITSs, and four had worked on six ITSs or more. Three people had worked on ITSs for three to six years and seven had worked on ITSs for seven years or more. One participant did not share any demographic data.

3. Results

3.1 Model architectures in current ITS development

Out of eleven participants, nine reported that the ITS he or she worked on most recently used a single learner model. Two reported using two learner models, and none reported using more than two constructs. The models participants used included representatives from five of the six architecture categories described in this paper. Example tracing was not represented. Note that the mention of a model type in this section *does* indicate current ITS research or development is using that architecture, but failure to mention a type does *not* indicate whether that architecture is in common use or not.

3.2 Development cost ratios

This section relates individual experiences with building different model types. The data are recent, since they represent participants' descriptions of the last project they completed. As elsewhere in this paper, cost is reported as a ratio reflecting the number of development person-hours spent to create one hour of individual instruction.

Table 3.2.1: Individual reports of *macro*adaptation models' development cost in relation to ITS teaching time.

Model Architecture	Cost Ratio
Overlay	24:1
Decision trees (Classifier)	30:1
Knowledge tracing	48:1
Model tracing	100:1
Overlay	667:1
Knowledge tracing	1375:1

Table 3.2.1 describes the cost of models supporting macroadaptation from six respondents who estimated both development time and instruction time. Table 3.2.2 gives the same information for microadaptation, as described by seven respondents. All participants in the study stated that they used microadaptation in their ITSs, and all but one used macroadaptation as well. Although macroadaptation costs were more variable, a two-tailed T-test did not find support in these responses for a significant difference between the cost of developing macroadaptation versus microadaptation.

Certain model types were represented more than once in the responses. Although these responses may come from different participants describing the same project, the likelihood is low because there was no instance when the details from one participant substantially matched another participant's response.

Table 3.2.2: Individual reports of *micro*adaptation models' development cost in relation to ITS teaching time.

Model Architecture	Cost Ratio
Overlay	24:1
Knowledge tracing	48:1
Behavior transition networks (Classifier)	50:1
Differential model (Overlay)	100:1
Constraint-based model	100:1
Buggy model	133:1
Knowledge tracing	450:1

Table 3.2.3 shows seven responses relating the cost of building an entire ITS, not just the learner model, to the hours of instruction provided. Each ITS is described by the model types the respondents used. The next section relates the cost of model development to the cost of system development.

Table 3.2.3: Individual reports of an *entire ITS's* development cost in relation to its teaching time, showing models used.

Model Architecture	Cost Ratio
Classifiers	250:1
Constraint-based model	333:1
Overlays	400:1
Knowledge tracing *	500:1
Model tracing and differential models	600:1
Knowledge tracing *	2000:1
Overlay and buggy models	5333:1

In Table 3.2.3, two respondents (marked with an asterisk) stated that they used knowledge tracing but did not affirm using model tracing. Since knowledge tracing refers to a way of using a second learner model in conjunction with a cognitive tutor, it may be that these ITSs also used model tracing.

3.3 Learner model cost as a percentage of ITS cost

Eight participants reported development cost estimates for both a tutoring system as a whole and its learner model. Costs in this section are absolute values, so some new responses can be used that did not appear in the previous section because they lacked instruction time estimates. Taken as an aggregate, these responses show how much of an ITS's cost goes toward building its learner model.

Responses indicated that, in general, a learner model accounts for about a third of the cost of an ITS, with a mean reported ratio of 33%, a median of 31%, and a standard deviation of 28 percentage points. The responses were overall consistent, so that dropping one low and one high outlier brought the standard deviation to 9 percentage points. The low outlier used an overlay model, and the high outlier used knowledge tracing.

4. Discussion

4.1 Interpretations

Although the responses gathered in this survey provide valuable anecdotal insights, there are too few responses to apply a detailed statistical analysis. However, individual responses suggest some interesting trends. One interesting fact is the high variability of cost estimates when more than one participant described the same model type. The large differences might be attributable to modeling tasks related to the architecture, such as learning to use a new model type, or unrelated, such as spending more time eliciting knowledge from subject-matter experts. Unfortunately, this study cannot determine how much of the variation in cost reports was attributable to the different model types.

Although combining the conflicting cost reports as an average might give a better view of the effort a model requires under many different circumstances, it would be misleading to aggregate such sparse data. Instead, it is more useful to use the most favorable estimate for each model type as a best-case scenario. Since there is no upper limit on the development effort anyone can expend on any model, examining the lowest or best case instead helps show whether it is at least possible to spend low amounts of time.

The best-case cost estimates for building a learner model alone cluster into two groups. One group of models has a cost ratio of 50:1 or lower, while the other group has a cost ratio between 100:1 and 133:1. The very high cost estimates in the results are not best-case scenarios because other participants reported lower estimates for the same model categories. The model types in the low-cost group include overlays, classifiers, and knowledge-tracing models (which are typically implemented with a Bayesian or overlay model). The model types that cost more include buggy models, constraint-based models, and the production-rule models in cognitive tutors. Considering best-case scenarios only, these model types cost between two and 5.5 times as much as the low-cost models.

Table 4.1.1: *Best-case scenario* model costs, as determined by finding the lowest cost ratio reported for each model category.

Model Architecture	Cost Ratio
Overlays	24:1
Classifiers	30:1
Knowledge tracing	48:1
Constraint-based model	100:1
Production-rule model (model tracing)	100:1
Buggy model	133:1

Estimating the cost of building a whole ITS, not just a learner model, makes values in this study comparable to published estimates of this figure. The costs of model-tracing tutors and constraint-based tutors reported in this study are approximately equal to figures published in the academic literature.

Using the reasoning discussed above in this section, the two whole-ITS cost ratios over 1000:1 in Table 3.2.3 do not represent best-case scenarios because there are lower cost estimates with the same model types. The remaining values in that table are all on the same order of magnitude and even the highest estimate, 600:1 for a model tracing cognitive tutor, was only 2.4 times as high as the lowest estimate. Although these estimates are quite close to each other, the responses do suggest that changing the learner model might halve or double the development time of the entire ITS.

The different responses in Table 3.2.3 also suggest an ordering of system development costs by learner model type. Using classifiers as learner models may lead to the fastest ITS development. This confirms intuitions that classifiers, as off-the-shelf tools, are easy to use and do not require publications about their development effort.

Surprisingly, tutoring systems using overlay models fell in the middle of the pack at best, despite the low cost of overlay models compared to other types in this study. However, this unexpected result may be due to the cost of knowledge elicitation on the two projects in question, rather than any costs directly associated with overlay models.

Considering whole-system costs, constraint-based systems are somewhat easier to develop than cognitive tutors, a conclusion which concurs with published anecdotes. The best-case costs of building a tutor with model-tracing or knowledge-tracing are higher than that of a constraint-based tutor, despite the fact that considering the learner model alone, constraints cost the same or more (see Table 4.1.1). A possible factor that might contribute to this difference is that constraint-based systems can work with less precise learner models, which might lead to less effort in creating specific hints and remediations for many different errors (Mitrovic et al., 2003). Cognitive tutors, with their model tracing and knowledge tracing algorithms, took the most effort of any ITS to build, confirming the intuition that led to constraint-based modeling and example tracing.

4.2 Limitations

Limitations of this study include a small population size, possible selection bias, and possible lack of consideration in forming estimates. Although the number of responses reported in this paper is comparable to the number of related publications from the academic community, that number does not yet reach levels that would allow a detailed statistical analysis. Furthermore, participants were not invited randomly, and invitees with certain characteristics may have been more or less likely to respond. Finally, ITS researchers who include development costs in publications can support their figures with careful records, while respondents in this study had to estimate costs after the fact. Because of these limitations, responses in this paper should be viewed as anecdotes rather than predictions of future performance. Although this study presents anecdotal evidence, it is still valuable input into choosing a learner model architecture if the limitations are understood.

5. Conclusion

This paper has presented anecdotal evidence concerning the development cost of learner models in ITSs. ITSs focus on personalization for every user, and this study showed that their learner models often account for about one third of their development cost. Different learner models have different costs to develop. In this study, eleven ITS practitioners from industry, academia, and military organizations shared their valuable experiences to provide anecdotal evidence about those costs.

The anecdotes in this paper, which align with the few published experiences previously available, suggest that certain learner models can be easier to build than others. Overlay models and classifiers used as learner models have the lowest development costs. With current authoring tools, constraint-based learner models are approximately as expensive to build as production-rule models. Buggy learner models are the most expensive to develop. The differences in model costs are also reflected in smaller but still noticeable differences in the cost of the entire ITS.

This study only addresses learner model development costs. It may be the case that more expensive learner models produce such good cognitive fidelity (Neches, Langley, & Klahr, 1987), effects on learning outcomes, or other benefits that they justify their cost or more. The authors of this paper are currently in the process of exploring this new data on model cost in relation to ITS benefits, that is, return on investment.

6. References

Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K. R. (2006). *The cognitive tutor authoring tools (CTAT): Preliminary evaluation of efficiency gains*. Paper presented at the 8th International Conference on Intelligent Tutoring Systems.

Anderson, J. R. (1993). *Rules of the Mind*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Arroyo, I., Woolf, B. P., & Beal, C. R. (2006). Addressing cognitive differences and gender during problem solving. *Technology, Instruction, Cognition & Learning, 4*, 31–63.

Baker, R. S. J., Corbett, A. T., & Aleven, V. (2008). *More Accurate Student Modeling Through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing*. Paper presented at the 9th International Conference on Intelligent Tutoring Systems.

Brown, J. S., & VanLehn, K. (1980). Repair theory: A generative theory of bugs in procedural skills. *Cognitive Science: A Multidisciplinary Journal, 4*(4), 379–426.

Burton, R. R. (1982). Diagnosing bugs in a simple procedural skill. In D. Sleeman & J. S. Brown (Eds.), *Intelligent Tutoring Systems* (pp. 157–183). London: Academic Press.

Burton, R. R., & Brown, J. S. (1976). *A tutoring and student modelling paradigm for gaming environments*. Paper presented at the SIGCSE-SIGCUE joint symposium on Computer science education.

Carbonell, J. (1970). AI in CAI: An artificial-intelligence approach to computer-assisted instruction. *IEEE Transactions on Man-Machine Systems, 11*(4), 190–202.

Carr, B. P. B. (1977). *Wusor II: A Computer Aided Instruction Program with Student Modelling Capabilities.* Massachusetts Institute of Technology, Cambridge, MA.

Castellano, M., Mastronardi, G., Di Giuseppe, G., & Dicensi, V. (2007). *Neural Techniques to Improve the Formative Evaluation Procedure in Intelligent Tutoring Systems*. Paper presented at the IEEE International Conference on Computational Intelligence for Measurement Systems and Applications.

Cha, H. J., Kim, Y. S., Park, S. H., Yoon, T. B., Jung, Y. M., & Lee, J. (2006). *Learning styles diagnosis based on user interface behaviors for the customization of learning interfaces in an intelligent tutoring system*. Paper presented at the 8th International Conference on Intelligent Tutoring Systems.

Conati, C., & Maclaren, H. (2005). *Data-driven refinement of a probabilistic model of user affect*. Paper presented at the 10th International Conference on User Modeling.

Conati, C., & Zhou, X. (2004). *Building and evaluating an intelligent pedagogical agent to improve the effectiveness of an educational game*. Paper presented at the 9th International Conference on Intelligent User Interfaces.

Corbett, A. T., & Bhatnagar, A. (1997). *Student Modeling in the ACT Programming Tutor: Adjusting a Procedural Learning Model With Declarative Knowledge*. Paper presented at the 6th International Conference on User Modeling. from um.org

Folsom-Kovarik, J. T., Schatz, S., & Nicholson, D. (in preparation). *Return on investment: A practical review of ITS student modeling techniques*. Orlando, FL: University of Central Florida.

Hatzilygeroudis, I., & Prentzas, J. (2004). Using a hybrid rule-based approach in developing an intelligent tutoring system with knowledge acquisition and update capabilities. *Expert Systems with Applications, 26*(4), 477–492.

Johnson, W. L. (1990). Understanding and debugging novice programs. *Artificial Intelligence, 42*, 51–97.

Kodaganallur, V., Weitz, R. R., & Rosenthal, D. (2005). A comparison of model-tracing and constraint-based intelligent tutoring paradigms. *International Journal of Artificial Intelligence in Education, 15*(2), 117–144.

Koedinger, K. R., Aleven, V., Heffernan, N., McLaren, B., & Hockenberry, M. (2004). *Opening the door to non-programmers: authoring intelligent tutor behavior by demonstration*. Paper presented at the 7th Annual Intelligent Tutoring Systems Conference.

Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education, 8*(1), 30–47.

Lee, C. S. (2007). Diagnostic, predictive and compositional modeling with data mining in integrated learning environments. *Computers & Education, 49*(3), 562–580.

Luckin, R., & du Boulay, B. (1999). Ecolab: The development and evaluation of a Vygotskian design framework. *International Journal of Artificial Intelligence in Education, 10*(2), 198–220.

Martin, B., & Mitrovic, A. (2002). *Automatic problem generation in constraint-based tutors*. Paper presented at the 6th International Conference on Intelligent Tutoring Systems.

Martin, B., Mitrovic, A., & Suraweera, P. (2008). ITS domain modelling with ontology. *Journal of Universal Computer Science, 14*(17), 2758–2776.

McQuiggan, S. W., Mott, B. W., & Lester, J. C. (2008). Modeling self-efficacy in intelligent tutoring systems: An inductive approach. *User Modeling and User-Adapted Interaction, 18*(1), 81–123.

Mitrovic, A., Koedinger, K. R., & Martin, B. (2003). *A comparative analysis of cognitive tutoring and constraint-based modeling*. Paper presented at the 9th International Conference on User Modeling.

Mitrovic, A., & Ohlsson, S. (1999). Evaluation of a constraint-based tutor for a database language. *International Journal of Artificial Intelligence in Education, 10*(3–4), 238–256.

Mitrovic, A., Suraweera, P., Martin, B., Zakharov, K., Milik, N., & Holland, J. (2006). *Authoring constraint-based tutors in ASPIRE*. Paper presented at the 8th International Conference on Intelligent Tutoring Systems.

Neches, R., Langley, P., & Klahr, D. (1987). Learning, development, and production systems. In D. Klahr, P. Langley & R. Neches (Eds.), *Production system models of learn-*

ing and development (pp. 1–53). Cambridge, MA: MIT Press.

Ohlsson, S. (1992). Constraint-based student modeling. *Journal of Artificial Intelligence and Education, 3*(4), 429–447.

Ott, S., Imoto, S., & Miyano, S. (2004). *Finding optimal models for small gene networks*. Paper presented at the Pacific Symposium on Biocomputing.

Payne, S. J., & Squibb, H. R. (1990). Algebra mal-rules and cognitive accounts of error. *Cognitive Science: A Multidisciplinary Journal, 14*(3), 445–481.

Razzaq, L., Patvarczki, J., Almeida, S. F., Vartak, M., Feng, M., Heffernan, N. T., et al. (2008). The ASSISTment Builder: Supporting the Life Cycle of ITS Content Creation. *IEEE Transactions on Learning Technologies, 2*(2), 157–166.

Snow, R. E., & Swanson, J. (1992). Instructional psychology: Aptitude, adaptation, and assessment. *Annual Review of Psychology, 43*(1), 583–626.

Stottler, R. H., Fu, D., Ramachandran, S., & Vinkavich, M. (2001). *Applying a Generic Intelligent Tutoring System (ITS) Authoring Tool to Specific Military Domains*. Paper presented at the Interservice/Industry Training, Simulation & Education Conference.

Suraweera, P., Mitrovic, A., & Martin, B. (2007). *Constraint Authoring System: an empirical evaluation*. Paper presented at the 13th International Conference on Artificial Intelligence in Education.

VanLehn, K. (1982). Bugs are not enough: Empirical studies of bugs, impasses and repairs in procedural skills. *Journal of Mathematical Behavior, 3*(2), 3–71.

Walonoski, J. A., & Heffernan, N. T. (2006). *Detection and analysis of off-task gaming behavior in intelligent tutoring systems*. Paper presented at the 8th International Conference on Intelligent Tutoring Systems.

Appendix: Questionnaire

Because of space restrictions, only the parts of the questionnaire that produced data used in this paper are reproduced in this appendix. However, a full version of the questionnaire is available in (Folsom-Kovarik et al., in preparation).

A recent ITS

Please describe the intelligent tutoring system (ITS) you worked on most recently that is ready, or nearly ready, to interact with students.

1. For the ITS you worked on most recently, approximately how many different student models did it use? *{No explicit student model, 1, 2, 3 or more modeling components}*

2. What student model type or modeling algorithm did the system use to SELECT MATERIAL to present? What did the system use to RESPOND TO ERRORS? If the system used more than one student model, please describe ONE model for each adaptation type.

Selecting or ordering material: *{Choose one or free response}*

Adapting corrections or hints: *{Choose one or free response}*

Did not use student modeling
Overlay model
Differential model
Perturbation model
Bug or bug-part library
Model tracing
Knowledge tracing
Example tracing
Other production-rule model
Constraint-based model
Case-based model
Finite-state automata
Behavior transition networks
Decision trees
Neural networks
Neurule system
Bayesian networks
Other (fill in below)

For the following questions, feel free to answer with an estimate, a range, or even an order of magnitude.

Please measure work in person-hours: each person working full-time for one week contributes about 40 person-hours, and one person working full-time for a year contributes about 2000 person-hours.

3. About how much work, measured in person-hours, did it take to create the ITS? How much of that time was spent working on the student models?

The whole ITS: *{Free response}*

The primary student model for MATERIAL SELECTION: *{Free response}*

The primary student model for HINTS AND FEEDBACK: *{Free response}*

4. Approximately how much additional time, measured in person-hours, was saved by reusing work from other projects?

The whole ITS: *{Free response}*

The primary student model for MATERIAL SELECTION: *{Free response}*

The primary student model for HINTS AND FEEDBACK: *{Free response}*

5. Did your team use any authoring tools to help build the ITS?

The whole ITS: *{Yes, No}*

The primary student model for MATERIAL SELECTION: *{Yes, No}*

The primary student model for HINTS AND FEEDBACK: *{Yes, No}*

6. (Optional) If so, which authoring tools did you use? *{Free response}*

7. When the project was finished, how many hours of instruction per student did the ITS provide? *{Free response}*

8. Are there any other comments you'd like to include about the student models in this ITS, how their design was determined, the model-building process, or anything else? *{Free response}*

Demographic information

As with all the questions in this survey, these questions are optional and you may leave any of them blank.

38. What type of organization do you work for? *{Industry, Government, Academic}*

39. Approximately how many adaptive education or training systems have you been involved with researching or creating? *{0, 1–2, 3–5, 6+}*

40. Approximately how long have you been involved with the research or development of adaptive technologies for education or training? *{N/A, 1–2 years, 3–6 years, 7+ years}*

Acknowledgements

This work is supported in part by the Office of Naval Research Grant N0001408C0186, the Next-generation Expeditionary Warfare Intelligent Training (NEW-IT) program. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the ONR or the US Government. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

Author Biographies

JEREMIAH T. FOLSOM-KOVARIK is a Research Assistant with the UCF-IST Applied Cognition and Training in Immersive Virtual Environments (ACTIVE) lab. He is a graduate student working toward a computer science PhD. His research with ACTIVE focuses on improving ITSs with novel user-system interactions and learner modeling opportunities.

SAE SCHATZ, Ph.D. is a Research Associate with the UCF-IST ACTIVE laboratory. Dr. Schatz performs applied research in scenario-based training, adaptive instruction, individual differences, and cultural modeling. She is experienced with both social and technological sciences, including human factors, human-computer interaction, educational psychology, computer programming, and instructional technology. Before joining ACTIVE, she served as an instructor in the UCF Digital Media Department, where she taught design philosophy, graphic arts, and web development.

DENISE NICHOLSON, Ph.D. is the Director of the UCF-IST ACTIVE laboratory. Her research focuses on human systems modeling, simulation and training includes virtual reality, human–agent collaboration, and adaptive human systems technologies for Department of Defense applications. She joined the university in 2005 with over 18 years of government service ranging from bench-level research at the Air Force Research Lab to leadership as the Deputy Director for Science and Technology at the U.S. Navy's NAVAIR Training Systems Division. She has authored more than 70 technical publications, and is coeditor of the three-volume work *The Handbook of Virtual Environments for Training and Education*, released November 2008.