In: B. Kennedy, D. Reitter, R. St. Amant (Eds.), *Proceedings of the 22nd Annual Conference on Behavior Representation in Modeling and Simulation*. Ottawa, Canada: BRIMS Society, 2013.

Declarative to Procedural Tutors: A Family of Cognitive Architecture-Based Tutors

Frank E. Ritter (frank.ritter@psu.edu)
College of IST
Kuo-Chuan (Martin) Yeh (myeh@psu.edu)
Department of Computer Science and Engineering
The Pennsylvania State University/University Park

Mark A. Cohen (mark.cohen@mcla.edu)
Department of Computer Science
Massachusetts College of Liberal Arts

Peter Weyhrauch (pweyhrauch@cra.com)
Charles River Analytics, Inc., Cambridge, MA

Jong W. Kim (jong.kim@ucf.edu)
Department of Psychology, U. of Central Florida

J. Nicholas Hobbs (hobbs@cs.umass.edu)
Department of Computer Science
U. of Massachusetts Amherst

ABSTRACT: *We have created a tutoring architecture, D2P (Declarative to Procedural), based on ACT-R's theory of knowledge and learning for declarative and procedural memory. D2P also draws on general theories of learning and a recent review by Kim, Ritter, and Koubek (in press). D2P provides multimedia instruction pages followed by questions, video, and/or simulations to test and proceduralise the declarative knowledge. It is implemented in XML, XML-based tools, Java, and COTS tools. We have created two D2P tutors. We have evaluated one, and it has a 1.48 effect size on learning how much to lead. We have found that modern tools help create a tutoring system; creating a tutoring architecture requires more than just the system (such as manuals and example systems); and the next steps are to test the theory of learning and use a cognitive architecture-based knowledge model to make the system adaptive.*

1. Introduction

To facilitate accelerated knowledge acquisition through repetitive practice, we have created a new tutoring architecture, D2P (Declarative to Procedural). D2P is based on the learning and memory theories in ACT-R (Anderson, 2007) and on other learning and cognitive science theories. It is focused on developing tutors for procedural skills that arise from declarative knowledge. These skills are sometimes found in schools but are more often found in training environments. To illustrate its use, and also as a way to develop the approach, we have created two tutors using D2P and evaluated the effectiveness of one tutor. D2P/MTT has a 1.48 effect size on learning how much to lead (how many points of aim).

After a review of previous tutoring systems and learning theories, we briefly present the architecture of D2P. Then, we present the two tutors and the evaluation of one of them. We conclude with future work and lessons learned.

2. Previous Tutoring Systems and Learning Theories

In this section, we review the basis of D2P tutors: previous intelligent tutoring systems (ITSs), learning theories, and some relevant theories from human-computer interaction (HCI) and cognitive science.

2.1 KRK Learning Theory: Declarative to Procedural

General theories of learning have been described in domains such as behavioral psychology (Fitts, 1964), cognitive psychology and cognitive science (e.g., Anderson, 1982; Rosenbloom & Newell, 1987), and cognitive engineering (e.g., Rasmussen, 1983). These theories suggest a consensus understanding that a learning process consists of a number of stages starting from acquiring declarative knowledge to forming procedural knowledge by practice, shown in Figure 1.

This learning process has been implemented in a cognitive architecture, ACT-R—it encodes facts of task knowledge, and, with practice, converts the acquired knowledge into a procedural form of knowledge based on both the activation mechanism of declarative chunks and the compilation mechanism of production rules (e.g., Anderson, 2007). One of the important implications of this learning theory is that forgetting and retention will vary across the three stages of learning (Kim, Ritter, & Koubek, in press).

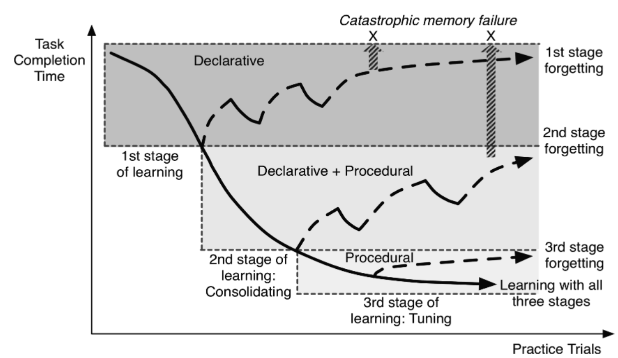

Figure 1. The KRK Theory of Skill Learning and Retention (Kim, Ritter, & Koubek, in press). X's represent catastrophic memory failure.

For the first stage of learning and forgetting, knowledge in declarative memory degrades with lack of use, perhaps catastrophically as indicated by the X's in Figure 1, leading to the inability to retrieve the memory and thus to perform the task. Decreased memory strength leads to increased response times and decreased retention and performance accuracy. In the second stage of learning, practice compiles declarative knowledge into procedural knowledge, and task knowledge is represented using a mix of declarative and procedural memory. With lack of use, the declarative knowledge is forgotten, leading to mistakes and missed steps. Procedural memory, on the other hand, is basically immune to decay. Soar and ACT-R (explicitly), and EPIC (implicitly) assume this, and it can be seen to be essentially true in some empirical studies (e.g., Stefanidis, Korndorffer, Markley, Sierra, & Scott, 2006). In the third stage, task knowledge is available in both declarative and procedural forms, but procedural knowledge predominantly drives performance.

This theory suggests that declarative knowledge has to be presented to the learner, that practice retrieving helps solidify declarative knowledge, and that practice using strong declarative memories will help form procedural memories. The D2P architecture is an attempt to test and apply this theory based on those cognitive architectures.

2.2 General tutors

In an unpublished review, Michael Qin (NSMRL) found several types of tutoring systems, including learning content management systems (CMSs), auxiliary tools, and self-contained authoring environments. We draw briefly on two of these categories and deeply on one.

Like CMSs, D2P tutors will have to have some content management support so that learners and instructors know how long the learner has spent with the tutor and what the learner has learned.

Qin found that there were many auxiliary tools used to prepare materials, including Powerpoint, drawing tools, and video editing tools. D2P will use these tools where needed and will not create or directly incorporate these tools. For example, in the tutors presented below we created storyboards initially in Powerpoint, which was easier than creating an outlining tool. (Since then, however, we have the ability to print out all the tutoring pages, which might ameliorate the use of Powerpoint in this way or provide a wireframe version.)

Self-contained authoring environments have several lessons. Two of the most prominent ones are Cognitive Tutor Authoring Tools (CTAT) and Generalized Intelligent Framework for Tutoring (GiFT), but Qin reviews over 60 others. There are even more, including Plato and other systems (Feurzeig, 2011; Murray, Blessing, & Ainsworth, 2003; Wenger, 1987).

CTAT (Aleven, McLaren, Sewall, & Koedinger, 2009) is an example of an ITS. An ITS is an application of AI to education, in which AI theories and algorithms are used to create a software application as a tutor. The richness of AI theory and complexity of creating such systems can make ITSs hard to adopt to different tutoring subjects. Authoring is another major hurdle to overcome for ITS end users, who are often educators, not AI experts. The CTAT research team is trying to make authoring a tutor easier and faster for the same or similar topics. To do so, CTAT provides tools (called widgets) for content authors in a drag-and-drop user interface to build the knowledge hierarchy of an expert solution. The goal is to provide an alternative authoring method to simplify the processes of creating tutors and to reduce the expertise required. Simplifying the process of creating tutors is a very important goal, however, CTAT seems difficult to use to us and its source code was not available to extend it.

CTAT uses modular design to separate system components that can be used by different user groups to build a tutor. With regard to the assessment, CTAT contains a component (DataShop) that records student-tutor interaction data for analysis, experiment, and display.

Research studies show that cognitive tutors built using CTAT or its underlying approach can improve students' performance significantly in the topics that are being trained (Anderson, Corbett, Koedinger, & Pelletier, 1995; Lesgold, 2001). CTAT tutors usually emphasize training students' problem solving ability and deep cognitive thinking skills with no time limit. Students can take as long as the system allows as long as they are on the correct solution path. In other words, these cognitive tutors are designed to improve performance through better problem solving skills.

GiFT is a collection modules that each focus on a different aspect of a tutoring system (Sottilare, Brawner, Goldberg, & Holden, 2012). These modules can be categorized into authoring, instruction, and analyzing. The goals of GiFT are to reduce the cost of using intelligent tutors in different tutoring domains, to easily use different pedagogical strategies, to integrate learner preferences and performances, and ultimately to use intelligent agents to make tutoring more effective.

GiFT is an attempt to solve several problems: tutoring systems are often specific to a domain, costly to build, and difficult to adapt to different tutoring subjects. Our architecture, D2P, has similar goals: to modularize components and to generalize the tutoring framework. Like GiFT, we will include a supporting environment to create the tutors and to run them.

CTAT and GiFT suggest that a problem or page-based approach can be appropriate, because they both use this approach. In addition, we will move the representations of what a learner has studied and mastered into a strong theory of task knowledge, that of the problem space computational model (PSCM, Newell, Yost, Laird, Rosenbloom, & Altmann, 1991), a level above rules but still a very concrete task analysis approach, implemented in the Herbal high level cognitive modeling language (Cohen, Ritter, & Haynes, 2010).

2.3 Related theories that Influence D2P

We are also mindful of Sweller's (1988) cognitive load theory, in that the tutor has to be easy to use so the learner can focus on the topic being learned. The example tutors we created using D2P (and in the manual including how to use notes), we introduce new technology elements gradually so that if there is a new interface, per se, to learn, it is a separate lesson. For example, in one of our example tutors the first lesson is to treat a hamster bite, which does not require much first aid knowledge, but allows the learner to learn about applying first aid using the interface. Therefore, the learner can focus on understanding the interface without also trying to apply new material learned from the tutor.

2.4 Lessons from Previous Systems

Our review suggests several lessons for creating a tutor system. Tutors should provide the declarative knowledge to be learned. In previously created tutoring systems, this knowledge has been presented through text, pictures, video, and simulations. D2P supports these.

Tutors should provide opportunities to practice the knowledge so that learners can proceduralise the knowledge. Without practice learners are unlikely to proceduralise their new declarative knowledge. Simple tutors will fail in this way. D2P should particularly emphasize creating multiple practice opportunities.

This need to provide practice sessions has several implications. The system should provide the ability for the learner to practice the declarative knowledge, through questions, text, multimedia, or simulation, or a combination. The questions and interaction with simulations created by the designer provide opportunities for learners to proceduralise knowledge.

While the tutoring system must support learners, we should explore supporting instructional designers more directly so that the new system will be easy to use by both its direct developers and also peripheral instructional designers. Creating and incorporating new educational material and questions must be easy for the instructional designer. Systems that are not easy to use do not transition easily out of the laboratory that created them.

3. The D2P Theory and Architecture

In this section, we describe our D2P architecture and the tools we built to use the architecture.

Figure 2 shows the D2P architecture. The instructional designer creates resources for the tutoring engine to use, including multimedia (outside D2P), XML tutor pages and questions (created within D2P using tools such as the page editor and question builder), and a task model (created in Herbal based on the PSCM, the higher structure in the Soar architecture). The D2P engine produces screens for the learner to interact with. These pages include text, video, pictures, questions, simulations, and a depiction of the PSCM task description. The time on task and question results are stored in a database.

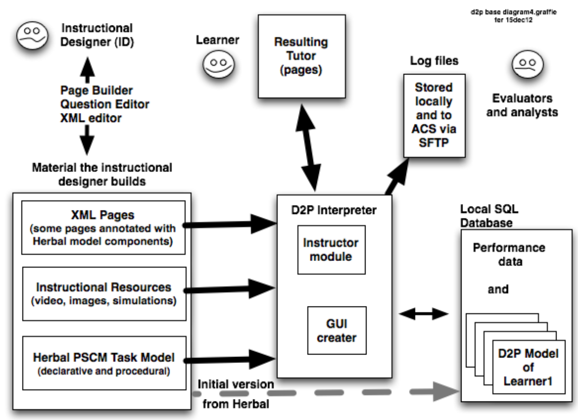

Figure 2. Overview of the D2P architecture.

The instructional designer's job has been simplified with the creation of a page editor and a question builder. These tools allow the instructional designer to create pages and questions graphically, without editing the XML representation. Pages and questions created using these tools can be reused across different tutors. The page editor can print the tutor pages to an HTML file for sharing and debugging, albeit with some inevitable simplifications (e.g., "Questions shown here", "simulation run here"). In addition, the question builder allows for questions to be imported and exported as CSV files. Three instructional designers have used these tools, and user manuals provide support (Hiam, Ritter, & Morgan, 2012).

4. Using the Architecture to Create Tutors

We have used D2P to create two tutors and are looking for other applications. We describe both tutors and an evaluation of one.

4.1 D2P/MTT

The Moving Target Tutor (D2P/MTT) is designed to teach people to shoot a moving target by teaching them to retrieve the correct Point of Aim (PoA) for hitting moving objects, which means how far to lead a target to account for elements such as how fast the target is moving, how fast the bullet travels, and which angle the target is moving. Determining the correct PoA is the result of combining the range, speed, and angle of a moving object relative to the observer. Quickly and correctly judging PoA is an important skill for any shooter who wants to hit a moving target.

To teach this skill, D2P/MTT contains roughly two to five hours of instructional materials, including text, still images, audio, videos, and quizzes. Learners use D2P/MTT to first learn the declarative knowledge necessary to determine the correct PoA, and then, through practice, the learner's declarative knowledge transforms into procedural knowledge, so the learner can both accurately and quickly determine the correct PoA.

To evaluate the effectiveness of D2P/MTT to proceduralise declarative knowledge in learners, we have data from 10 PSU students 18 to 24 years old. All participants used D2P/MTT (v. 0.4.1) in a quiet room on a Windows XP laptop.

Participants first finished a pre-test asking them to judge angle, range, speed, or points-of-aim to hit a moving target, using 24 videos showing targets running or walking (varied target speed), at 25, 50, or 100 yards (varied range), and at 45 or 90 degree angles (varied angle). This is summarized in Table 1. These questions were presented in a fixed order without feedback to minimize the potential for learning during the pre-test, and presented in a different order in the post-test. Questions were shuffled among categories.

Table 1: Number of questions for each category

Category	Number of Questions
Angle (45 and 90 degree)	4
Range (25, 50, and 100 yards)	6
Speed (5 and 10 mph)	4
Points-of-aim	10

In the D2P/MTT pre-test, and later in the tutor, participants would read the question, watch the video, enter an answer, and then advance to the next question. All user interactions were recorded using the RUI keystroke and mouse logger (Kukreja, Stevenson, & Ritter, 2006).

After the pre-test, participants completed the D2P/MTT training (approximately 100 pages) and practice questions (46 questions) with feedback telling them whether their answers were right or wrong and what the right answers were. After the training sessions, participants were tested using the post-test. The pre-test and post-test questions were in different orders. Table 2 shows the results. Further details are available (Yeh & Ritter, 2012).

Table 2. Descriptive Statistics for the Study

	Pre (SD)	Post (SD)	T(9)	P
PoA Questions correct out of 10	2.10 (1.66)	4.70 (1.77)	3.34	.0086
Video Test Completion time (sec.)	279.6 (47.4)	189.7 (30.3)	5.34	.00047

The Cohen's effect size (d) is 1.48, which shows D2P/MTT is effective (it is the 5th highest out of more than 50 reported by VanLehn, 2011), but could be improved. A revised tutor with more material and more practice has been created and is being tested this spring at Penn State and at Quantico.

4.2 D2P/CLS

We also developed a tutor to teach first-aid topics for a Marine qualification called Combat Lifesaver—D2P/CLS consists of over 150 instructional pages. In addition to text, images, questions, and multimedia, it includes a simulation with four interactive content areas, as shown in Figure 3. On the left side of the simulation the tutee has access to two first aid kits. (One is the casualty's and the other the learner's.) At the bottom of the simulation window is a tool bar of available actions the tutee can perform to treat the casualty. Status bars indicating the patient's blood volume level and oxygen level are at the top. Finally, the tutee can interact with an interactive image of the casualty to query the casualty's status and to provide care.

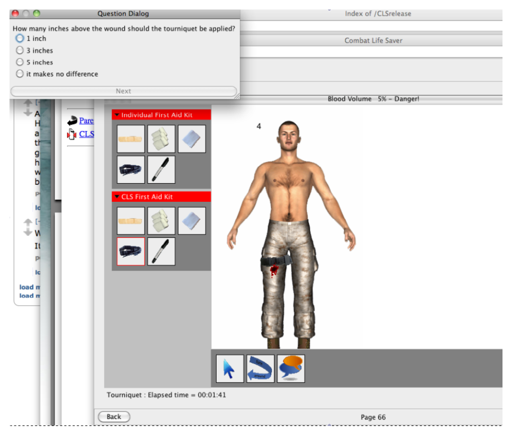

Figure 3. The simulation and questions in the simulation in the D2P/CLS tutor. The two first aid kits, the action bar, and the main content area are shown.

When the tutoring environment launches the simulation, it loads a scenario script file defining properties including the maximum scenario duration, which status bars should be visible, equipment that should be available in the first aid kits, and the casualty's wounds. The wound properties specify factors such as the location of the wound, the bleed rate, and how likely it is that the wound will remain stable after it is treated. The script file keeps the simulation modular.

Another method of customizing the simulation allows designers to plug in custom Java classes that alter or define new first aid equipment, medical treatment actions, and wound types. This allows designers with programming experience to design new components for inclusion in novel scenarios.

Like D2P/MTT, the D2P/CLS tutor uses questions that were created by the instructional designer using the question builder. These questions can be inserted either as a declarative test or into a simulation. The question builder allows for both of these styles of questions to exist together, and later to be added into the content by only specifying the question database and the category of question requested. This separation allows the designer to reuse questions within a tutor or across multiple tutors, resulting in less work for the instructional designer.

While the simulation is running, D2P can query the status of the current scenario or halt the scenario. In addition, the tutor can register to be notified if the simulation terminates on its own. When the simulation exits, an exit code is returned to the tutor specifying how successful the tutee was at treating the patient. In this case, the exit code reflects the blood level and oxygen level of the patient, as well as the elapsed time.

4.3 Summary

D2P/MTT and D2P/CLS illustrate several aspects of D2P. They include several types of multimedia and a simulation, and they have driven the development of the question builder and the use of questions in simulations, and their needs have helped extend the page builder. D2P/MTT has been tested and has a 1.485 effect size.

5. Discussion and Conclusions

This paper has discussed briefly the D2P system and some of its philosophy, development approach, and two example systems. It has noted how an early version of one of the systems has been tested and led to greater learning, better than most ITS reported in a recent review (VanLehn, 2011), and yet created by relatively novice instructional designers. In this discussion we note some lessons we have learned, and future work.

5.1 COTS Systems and Public Libraries are Helpful

We have found that tutor development can now be supported by existing software tools and libraries that should not be reinvented, and even does not need to be integrated with the tutoring software (i.e., the tutoring system does not have to edit video within itself). For example, we build tutor prototypes using commercial off-the-shelf (COTS) tools, such as PowerPoint or balsamiq®. Such prototypes can be rapidly created, reviewed, and revised before building in D2P (which we did for both tutors). Further, using readily available and affordable COTS tools for multimedia preparation, including image and video editors, allows instructional designers to use tools they are already familiar with, making D2P easier to use. Finally, the modular design of D2P allows it to take advantage of existing software tools and libraries, such as video playback, data logging, or simulation systems that are available in Java and using XML. For example, allowing a D2P tutor to play movies using a Java library or connecting to an existing simulation, such as VBS2, provides an opportunity to create a tutoring experience that is richer than if this functionality had to be created (again) within D2P.

5.2 The Use of Modular Design

Recent tutoring systems seem to agree on making a tutoring system a collection of modules so that the entire system is more flexible and easier to maintain and use. Our approach is the same. We separate the pedagogical module (the XML representation and question builder database), the instructional module (the Java user program), the authoring module (the page builder and question builder), and the analytical module (the logs in a database). These modules can be changed or replaced without affecting others. This design also allows us to add new modules or have a library of modules for different domains, different pedagogical strategies, etc.

5.3 Towards an Intelligent Tutoring System

Using a cognitive architecture within a system is not easy. We have found that we have had to create a lot of infrastructure to test our theory of learning. We have created a tutoring engine, tools for creating content, and manuals and examples to demonstrate where we are and to help new instructional designers.

The next steps are to fundamentally test the theory of learning, that practice after repeated declarative practice leads to faster procedural learning, and that we can use Herbal PSCM constructs to choose pages to instruct.

References

Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological Review, 89*, 369-406.

Anderson, J. R. (2007). *How can the human mind exist in the physical universe?* New York, NY: Oxford University Press.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *Journal of the Learning Sciences, 4*(2), 167-207.

Cohen, M. A., Ritter, F. E., & Haynes, S. R. (2010). Applying software engineering to agent development. *AI Magazine, 31*(2), 25-44.

Feurzeig, W. (2011). Educational technology at BBN. In D. Walden & R. Nickerson (Eds.), *A culture of innovation*. East Sandwich, MA: Waterside Publishing.

Fitts, P. M. (1964). Perceptual-motor skill learning. In A. W. Melton (Ed.), *Categories of human learning* (pp. 243-285). New York: Academic Press.

Hiam, J. W., Ritter, F. E., & Morgan, J. H. (2012). *Declarative To Procedural Manual* (Tech. Report No. 2012-1): Applied Cognitive Science Lab, College of Information Sciences and Technology, Penn State.

Kim, J. W., Ritter, F. E., & Koubek, R. J. (in press). An integrated theory for improved skill acquisition and retention in the three stages of learning. *Theoretical Issues in Ergonomics Science*.

Kukreja, U., Stevenson, W. E., & Ritter, F. E. (2006). RUI—Recording User Input from interfaces under Windows and Mac OS X. *Behavior Research Methods, 38*(4), 656–659.

Lesgold, A. M. (2001). The nature and methods of learning by doing. *American Psychologist, 56*(11), 964-973.

Murray, T., Blessing, S., & Ainsworth, S. (2003). *Authoring tools for advanced technology learning environments*. Dordrecht, The Netherlands: Kluwer.

Newell, A., Yost, G. R., Laird, J. E., Rosenbloom, P. S., & Altmann, E. (1991). Formulating the problem space computational model. In R. F. Rashid (Ed.), *Carnegie Mellon Computer Science: A 25-Year commemorative* (pp. 255-293). Reading, MA: ACM-Press (Addison-Wesley).

Rasmussen, J. (1983). Skills, rules, knowledge: Signals, signs and symbols and other distinctions in human performance models. *IEEE Transactions: Systems, Man, & Cybernetics, SMC-13*, 257-267.

Rosenbloom, P. S., & Newell, A. (1987). Learning by chunking, a production system model of practice. In D. Klahr, P. Langley & R. Neches (Eds.), *Production system models of learning and development* (pp. 221-286). Cambridge, MA: MIT Press.

Sottilare, R. A., Brawner, K. W., Goldberg, B. S., & Holden, H. K. (2012). *The Generalized Intelligent Framework for Tutoring (GIFT)*: U.S. Army Research Laboratory—Human Research & Engineering Directorate (ARL-HRED) SFC Paul Ray Smith Simulation & Training Technology Center Learning in Intelligent Tutoring Environments (LITE) Laboratory.

Stefanidis, D., Korndorffer, J. R., Markley, S., Sierra, R., & Scott, D. J. (2006). Proficiency maintenance: Impact of ongoing simulator training on laparoscopic skill retention. *Journal of the American College of Surgeons, 202*, 599-603.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science, 12*, 257-285.

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist, 46*(4), 197–221.

Wenger, E. (1987). *Artificial intelligence and tutoring systems: Computational and cognitive approaches to the communication of knowledge*. San Francisco, CA: Morgan Kaufman.

Yeh, K.-C., & Ritter, F. E. (2012). *An initial evaluation of the D2P/MTT, a computer-based, Declarative to Procedural (D2P) theory driven moving target tutor* (Tech. Report No. ACS 2012-1). Applied Cognitive Science Lab, College of Information Sciences and Technology, Penn State.

Acknowledgements

This work was supported by ONR (N00014-11-1-0275, N00014-10-C-0281, N00014-10-1-0401), and partially sponsored by the U.S. Army Research Laboratory - Human Research & Engineering Directorate (ARL-HRED, W911QX-12-C-0063). Statements and opinions expressed in this paper do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred. We thank Wally Feurzeig, Ray Pursel, Jack Sparks, Henry Warren, Paul Andre (MCWL), and James Niehaus for comments and insights. Jeremiah Hiam, Jon Morgan, and Stephanie Harvill provided comments and helped created the tutors.

Author Biographies

FRANK RITTER is on the faculty of the College of IST, an interdisciplinary academic unit at Penn State to study how people process information using technology. He edits the *Oxford Series on Cognitive Models and Architectures* and is an editorial board member of *Human Factors, Cognitive Systems Research,* and *IEEE Transactions on Human-Machine Systems.*

KUO-CHUAN (MARTIN) YEH is an assistant professor in the Department of Computer Science and Engineering at Penn State. His research interests include applying technology to enhance learning performance and experience, particular through mobile devices, and HCI.

MARK COHEN is an Associate Professor of CS at the Massachusetts College of Liberal Arts. For the past eight years he has been developing software designed to help subject matter experts develop cognitive models and tutoring systems quickly.

PETER WEYHRAUCH is a Principal Scientist at Charles River Analytics, Inc, a developer of intelligent software solutions. His professional interests include AI, models of human behaviour and computational narrative technology. He is working in real-time interactive agents for entertainment and training, intelligent tutoring, preventing skill decay, and surgical and medical training.

JONG WOOK KIM is research faculty in Psychology at the University of Central Florida. His research pursuit is to improve cognitive systems supporting optimized user performance. His recent research, sponsored by ONR, has investigated skill learning and forgetting, and developed a theory of skill retention that is being applied to tutoring systems.

J. NICHOLAS HOBBS is a grad student in CS at the U. of Massachusetts Amherst. His research interests focus on understanding memory systems.