Preface

Robert A. Sottilare[1], Arthur C. Graesser[2], Xiangen Hu[2], and Heather Holden[1], Eds.

U.S. Army Research Laboratory - Human Research and Engineering Directorate[1]

University of Memphis Institute for Intelligent Systems[2]

This book is the first in a planned series of books that examine key topics (e.g., learner modeling, instructional strategies, authoring, domain modeling, learning effect, and team tutoring) in intelligent tutoring system (ITS) design through the lens of the Generalized Intelligent Framework for Tutoring (GIFT; Sottilare, Brawner, Goldberg, and Holden, 2012), a modular, service-oriented architecture created to develop standards for authoring, managing instruction, and analyzing the effect of ITS technologies.

This preface introduces tutoring functions, provides learner modeling examples, and examines the motivation for standards for the design, authoring, instruction, and analysis functions within ITSs. Next, we introduce GIFT design principles, and finally, we discuss how readers might use this book as a design tool. We begin by examining the concept of learner modeling.

Learner modeling (also known as student modeling or user modeling) is one of the major components of ITS. Learner modeling is a key to the computer-based tutor's understanding of the learner. Comprehensive, real-time modeling of the learner is a critical element in the design and development of truly adaptive tutoring systems that can tailor tutoring experiences to the needs of the individual learner and teams of learners.

It is generally accepted that an ITS has four major components (Elson-Cook, 1993; Nkambou, Mizoguchi & Bourdeau, 2010; Graesser, Conley & Olney, 2012; Psotka & Mutter, 2008; Sleeman & Brown, 1982; VanLehn, 2006; Woolf, 2009): The domain model, the student model, the tutoring model, and the user-interface model. GIFT similarly adopts this four-part distinction, but with slightly different corresponding labels (domain module, learner module, pedagogical module, and tutor-user interface) and the addition of the sensor module, which can be viewed as an expansion of the user interface.

(1) The domain model contains the set of skills, knowledge, and strategies of the topic being tutored. It normally contains the ideal expert knowledge and also the bugs, mal-rules, and misconceptions that students periodically exhibit.

(2) The learner model consists of the cognitive, affective, motivational, and other psychological states that evolve during the course of learning. It is often viewed as an overlay (subset) of the domain model, which changes over the course of tutoring. For example, "knowledge tracing" tracks the learner's progress from problem to problem and builds a profile of strengths and weaknesses relative to the domain model (Anderson, Corbett, Koedinger & Pelletier, 1995). An ITS may also consider psychological states outside of the domain model that need to be considered as parameters to guide tutoring.

(3) The tutor model (also known as the pedagogical model or the instructional model) takes the domain and learner models as input and selects tutoring strategies, steps, and actions on what the tutor should do next in the exchange. In mixed-initiative systems, the learners may also take actions, ask questions, or request help (Aleven, McClaren, Roll & Koedinger, 2006; Rus & Graesser, 2009), but the ITS always needs to be ready to decide "what to do next" at any point and this is determined by a tutoring model that captures the researchers' pedagogical theories.

(4) The user interface interprets the learner's contributions through various input media (speech, typing, clicking) and produces output in different media (text, diagrams, animations, agents). In addition to the conventional human-computer interface features, some recent systems have incorporated natural language interaction (Graesser et al., 2012; Johnson & Valente, 2008), speech recognition (D'Mello, Graesser & King, 2010; Litman, 2013), and the sensing of learner emotions (Baker, D'Mello, Rodrigo & Graesser, 2010; D'Mello & Graesser, 2010; Goldberg, Sottilare, Brawner, Holden, 2011).

The designers of the learner model need to decide what content, fields, variables, and parameters need to be included in the representation. The representation needs to be complete with respect to handling the distinctions made in the domain model, tutoring model, and user interface. Such representations vary in grain size, reflecting the complexity of the ITS. There is a comparatively small number of distinctions made in conventional computer-based training (O'Neil & Perez, 2003). For example, a simple system would just keep track of whether the learner has mastered (yes versus no) a set of N learning objects in the curriculum and the objects in a curriculum would be ordered theoretically, perhaps from simple to complex or along a prerequisite ladder (Gagne, 1985). The tutoring module would select the next learning object that the learner has not mastered and places that as lowest/earliest in the ordering. This kind of simple system may go a long way. However, ITSs presume to go a large step further in grain size and adaptability. Some of these ITSs are listed below, but there are many others that can be discussed at the workshop. One foundational question is whether the increased grain size and adaptability has an incremental return on investment with respect to learning gains.

Learner Modeling Examples

The following sections briefly describe four examples of learner modeling that have been developed and tested in contemporary ITSs.

Knowledge Tracing in the Cognitive Tutor

This approach to learner modeling tracks the learner's progress from problem to problem and builds a profile of strengths and weaknesses relative to the production rules (Anderson et al., 1995). A production rule is an "IF<state>THEN<action>" expression that specifies that a particular action, step, or cognitive event occurs in a particular state of the task or cognition. Information from knowledge tracing can be presented as a *skillometer*, a visual graph of the learner's success in each of the monitored skills related to solving problems in a step by step fashion. The skillometer is updated as the learner performs correct actions, commits errors, and requests a hint. Step-by-step knowledge tracing is incorporated in a number of tutors in the Pittsburgh Science of Learning Center (Aleven et al., 2006; Anderson et al., 1995; Heffernan, Koedinger & Razzaq, 2008; Ritter, Anderson, Koedinger & Corbett, 2007; VanLehn, 2006).

Constraint-based Modeling

In constraint-based tutors, a good solution is represented as a declarative structure and the learner's actions are compared with these constraints (Mitrovic, Martin & Suraweera, 2007; Ohlson, 1992). Each constraint is a declarative statement composed of a relevance condition (R) and a satisfaction condition (S). The relevance condition specifies when the constraint is relevant and only in these conditions is the state constraint meaningful. The satisfaction condition specifies whether the state constraint has been violated. A relevant, satisfied state constraint corresponds to an aspect of the correct solution. A relevant, unsatisfied state constraint indicates a flaw in the solution. Learner modeling is tracked by considering what constraints are followed as learners solve problems. Successful constraint-based tutors include Structured Query Language (SQL) tutor, *Knowledge-based Entity Relationship Modeling Intelligent Tutor* (KERMIT), and Addison-Wesley's *Database Place* (Mitrovic, Martin & Suraweera, 2007).

Knowledge Space Models

Knowledge space modeling underlies the Assessment and Learning in Knowledge Spaces (ALEKS) mathematics tutor (Doignon & Falmagne, 1999; Hu et al., 2012). The domain model of knowledge space theory is a large number of possible knowledge states on a topic, whereas the learner model is a record of which of the knowledge states are mastered, essentially a fine-grained overlay model. A learner's competence is reflected in the types of problems that the learner is capable of solving (among 250–350 problems), given the profile of knowledge states mastered among millions of possible states. Bayesian statistics are used to select the next problem to work on that is sensitive to the learner's competence by filling in deficits and correcting misconceptions. If a learner solves the next problem correctly, then each knowledge state containing that problem incrementally increases in probability; if the learner answers incorrectly, then the knowledge states are decreased in probability. Categories of skills are represented in a pie chart that reflects the competence of the learner.

Expectation and Misconception Tailored Dialogue

This type of learner modeling is typical for ITSs that help learners learn by holding a conversation in natural language, such as *AutoTutor* or *Why-Atlas* (Graesser et al., 2012; VanLehn et al., 2007). An answer to a question is a set sentence-like expectation (good answer), but the tutor also anticipates the learner articulating misconceptions (errors). An expectation or misconception is scored as being expressed by a learner if the learner articulates it in natural language with a high enough semantic match. Semantic matches can be assessed by a number of methods in computational linguistics, such as content word overlap, latent semantic analysis, regular expressions, semantic entailment, or Bayesian statistics (Cai et al., 2011; Graesser et al., 2007; Rus et al., 2009; VanLehn et al., 2007). When the total set of problems is considered, there is a universal set of expectations (called principles or facets) and misconceptions that are relevant to the various problems. These principles can be tracked over problems and guide the selection of the next problem to work on.

Motivations for Intelligent Tutoring System Standards

An emphasis on self-regulated learning has highlighted a requirement for point-of-need training in environments where human tutors are either unavailable or impractical. ITSs have been shown to be as effective as expert human tutors (VanLehn, 2011) in one-to-one tutoring in well-defined domains (e.g., mathematics or physics) and significantly better than traditional classroom training environments. ITSs have demonstrated significant promise, but fifty years of research have been unsuccessful in making ITSs ubiquitous in military training or the tool of choice in our educational system. Why?

The availability and use of ITSs have been constrained by their high development costs, their limited reuse, a lack of standards, and their inadequate adaptability to the needs of learners (Picard, 2006). Their application to military domains is further hampered by the complex and often ill-defined environments in which our military operates today. ITSs are often built as domain-specific, unique, one-of-a-kind, largely domain-dependent solutions focused on a single pedagogical strategy (e.g., model tracing or constraint-based approaches) when complex learning domains may require novel or hybrid approaches. Therefore, a modular ITS framework and standards are needed to enhance reuse, support authoring, optimize instructional strategies, and lower the cost and skillset needed for users to adopt ITS solutions for training and education. It was out of this need that the idea for GIFT arose.

GIFT has three primary functions: authoring, instructional management, and analysis. First, it is a framework for authoring new ITS components, methods, strategies, and whole tutoring systems. Second, GIFT is an instructional manager that integrates selected tutoring principals and strategies for use in ITSs. Finally, GIFT is an experimental testbed to analyze the effectiveness and impact of ITS components, tools, and methods. GIFT is based on a learner-centric approach with the goal of improving linkages in the adaptive tutoring learning effect chain (Figure P-1).

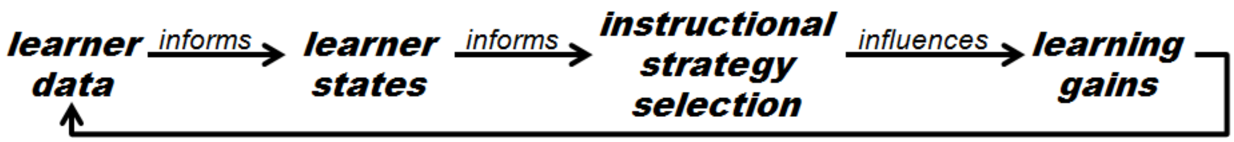

Figure P-1. Adaptive Tutoring Learning Effect Chain (Sottilare, 2012)

A deeper understanding of the learner's behaviors, traits, and preferences (learner data) collected through performance, physiological and behavioral sensors, and surveys will allow for more accurate evaluation of the learner's states (e.g., engagement level, confusion, frustration), which will result in a better and more persistent model of the learner. To enhance the adaptability of the ITS, methods are needed to accurately classify learner states (e.g., cognitive, affective, psychomotor, social) and select optimal instructional strategies given the learner's existing states. A more comprehensive learner model will allow the ITS to adapt more appropriately to address the learner's needs by changing the instructional strategy (e.g., content, flow, or feedback). An instructional strategy that is better aligned to the learner's needs is more likely to positively influence their learning gains. It is with the goal of optimized learning gains in mind that the design principles for GIFT were formulated.

GIFT Design Principles

The methodology for developing a modular, computer-based tutoring framework for training and education considered major design goals, anticipated uses, and applications. The design process also looked at enhancing one-to-one (individual) and one-to-many (collective or team) tutoring experiences beyond the state of practice for ITSs today. A significant focus of the GIFT design was on domain-dependent elements in the domain module. This was done to allow large-scale reuse of the remaining GIFT modules across different training domains and thereby reduce the development costs for ITSs.

One design principle adopted in GIFT is that each module should be capable of gathering information from other modules according to the design specification. Designing to this principle resulted in standard message sets and message transmission rules (i.e., request-driven, event-driven, or periodic transmissions). For instance, the pedagogical module is capable of receiving information from the learner module to develop courses of action for future instructional content to be displayed, manage flow and challenge level, and select appropriate feedback. Changes to the learner's state (e.g., engagement, motivation, or affect) trigger messages to the pedagogical module, which then recommends general courses of action (e.g., ask a question or prompt the learner for more information) to the domain module, which provides a domain-specific intervention (e.g., what is the next step?).

Another design principle adopted within GIFT is the separation of content from the executable code (Patil & Abraham, 2010). Data and data structures are placed within models and libraries, while software processes are programmed into interoperable modules. Efficiency and effectiveness goals (e.g., accelerated learning and enhanced retention) were considered to address the time available for military training and the renewed emphasis on self-regulated learning. An outgrowth of this emphasis on efficiency and effectiveness led Dr. Sottilare to seek external collaboration and guidance. In 2012, U.S. Army Research Laboratory (ARL) with the University of Memphis developed advisory boards of senior tutoring system scientists from academia and government to influence the GIFT design goals moving forward. An advisory board for learner modeling was completed in September 2012, and future boards are planned for instructional strategy design, authoring and expert modeling, learning effect evaluations, and domain modeling.

Design Goals and Anticipated Uses

GIFT may be used as any of the following:

1. An architectural framework with modular, interchangeable elements and defined relationships
2. A set of specifications to guide ITS development
3. A set of exemplars instantiating GIFT to support authoring and ease-of-use
4. A technical platform or testbed for guiding the development of concrete systems

These use cases have been distilled down into the three primary functional areas, or *constructs*: authoring, instructional management, and analysis. Discussed below are the purposes, associated design goals, and anticipated uses for each of the GIFT constructs.

GIFT Authoring Construct

The purpose of the GIFT authoring construct is to provide technology (tools and methods) to make it affordable and easier to build ITSs and ITS components. Toward this end, a set of extensible markup language (XML) configuration tools continues to be developed to allow for data-driven changes to the design and implementation of GIFT-generated ITSs. The design goals for the GIFT authoring construct have been adapted from Murray (1999, 2003) and Sottilare & Gilbert (2011). The GIFT authoring design goals are as follow:

- Decrease the effort (time, cost, and/or other resources) for authoring and analyzing ITSs by automating authoring processes, developing authoring tools and methods, and developing standards to promote reuse.
- Decrease the skill threshold by tailoring tools for specific disciplines (e.g., instructional designers, training developers, and trainers) to author, analyze, and employ ITS technologies.
- Provide tools to aid designers/authors/trainers/researchers in organizing their knowledge.
- Support (structure, recommend, or enforce) good design principles in pedagogy through user interfaces, and other interactions.
- Enable rapid prototyping of ITSs to allow for rapid design/evaluation cycles of prototype capabilities.
- Employ standards to support rapid integration of external training/tutoring environments (e.g., simulators, serious games, slide presentations, transmedia narratives, and other interactive multimedia).
- Develop/exploit common tools and user interfaces to adapt ITS design through data-driven means.
- Promote reuse through domain-independent modules and data structures.
- Leverage open-source solutions to reduce ITS development and sustainment costs.

- Develop interfaces/gateways to widely used commercial and academics tools (e.g., games, sensors, toolkits, virtual humans).

As a user-centric architecture, anticipated uses for GIFT authoring tools are driven largely by the anticipated users, which include learners, domain experts, instructional system designers, training and tutoring system developers, trainers and teachers, and researchers. In addition to user models and graphical user interfaces, GIFT authoring tools include domain-specific knowledge configuration tools, instructional strategy development tools, and a compiler to generate executable ITSs from GIFT components in a variety of formats (e.g., PC, Android, and IPad).

Within GIFT, domain-specific knowledge configuration tools permit authoring of new knowledge elements or reusing existing (stored) knowledge elements. Domain knowledge elements include learning objectives, media, task descriptions, task conditions, standards and measures of success, common misconceptions, feedback library, and a question library, which are informed by instructional system design principles that, in turn inform concept maps for lessons and whole courses. The task descriptions, task conditions, standards and measures of success, and common misconceptions may be informed by an expert or ideal learner model derived through a task analysis of the behaviors of a highly skilled user. ARL is investigating techniques to automate this expert model development process to reduce the time and cost of developing ITSs. In addition to feedback and questions, supplementary tools are anticipated to author explanations, summaries, examples, analogies, hints, and prompts in support of GIFT's instructional management construct.

GIFT Instructional Management Construct

The purpose of the GIFT instructional management construct is to integrate pedagogical best practices in GIFT-generated ITSs. The modularity of GIFT will also allow GIFT users to extract pedagogical models for use in tutoring/training systems that are not GIFT-generated. GIFT users may also integrate pedagogical models, instructional strategies, or instructional tactics from other tutoring systems into GIFT. The design goals for the GIFT instructional management construct are the following:

- Support ITS instruction for individuals and small teams in local and geographically distributed training environments (e.g., mobile training), and in both well-defined and ill-defined learning domains.

- Provide for comprehensive learner models that incorporate learner states, traits, demographics, and historical data (e.g., performance) to inform ITS decisions to adapt training/tutoring.

- Support low-cost, unobtrusive (passive) methods to sense learner behaviors and physiological measures and use these data along with instructional context to inform models to classify (in near real time) the learner's states (e.g., cognitive and affective).

- Support both macro-adaptive strategies (adaptation based on pre-training learner traits) and micro-adaptive instructional strategies and tactics (adaptation based learner states and state changes during training).

- Support the consideration of individual differences where they have empirically been documented to be significant influencers of learning outcomes (e.g., knowledge or skill acquisition, retention, and performance).

- Support adaptation (e.g., pace, flow, and challenge level) of the instruction based the domain and learning class (e.g., cognitive learning, affective learning, psychomotor learning, social learning).

- Model appropriate instructional strategies and tactics of expert human tutors to develop a comprehensive pedagogical model.

To support the development of optimized instructional strategies and tactics, GIFT is heavily grounded in learning theory, tutoring theory, and motivational theory. Learning theory applied in GIFT includes cognitive learning (Anderson & Krathwohl, 2001), affective learning (Krathwohl, Bloom, and Masia, 1964; Goleman, 1995), psychomotor learning (Simpson, 1972), and social learning (Sottilare, Holden, Brawner, and Goldberg, 2011; Soller, 2001). Aligning with our goal to model expert human tutors, GIFT considers the INSPIRE model of tutoring success (Lepper, Drake, and O'Donnell-Johnson, 1997) and the tutoring process defined by Person, Kreuz, Zwaan, and Graesser (1995) in the development of GIFT instructional strategies and tactics.

INSPIRE is an acronym that highlights the seven critical characteristics of successful tutors: Intelligent, Nurturant, Socratic, Progressive, Indirect, Reflective, and Encouraging. Graesser & Person's (1994) tutoring process includes a tutor-learner interchange where the tutor asks a question, the learner answers the question, the tutor gives feedback on the answer, then the tutor and learner collaboratively improve the quality of (or embellish) the answer. Finally, the tutor evaluates learner's understanding of the answer.

As a learner-centric architecture, anticipated uses for GIFT instructional management capabilities include both automated instruction and blended instruction, where human tutors/teachers/trainers use GIFT to support their curriculum objectives. If its design goals are realized, it is anticipated that GIFT will be widely used beyond military training contexts as GIFT users expand the number and type of learning domains and resulting ITS generated using GIFT.

GIFT Analysis Construct

The purpose of the GIFT analysis construct is to allow ITS researchers to experimentally assess and evaluate ITS technologies (ITS components, tools, and methods). The design goals for the GIFT analysis construct are the following:

- Support the conduct of formative assessments to improve learning

- Support summative evaluations to gauge the effect of technologies on learning

- Support assessment of ITS processes to understand how learning is progressing throughout the tutoring process

- Support evaluation of resulting learning versus stated learning objectives

- Provide diagnostics to identify areas for improvement within ITS processes

- Support the ability to comparatively evaluate ITS technologies against traditional tutoring or classroom teaching methods

- Develop a testbed methodology to support assessments and evaluations (Figure P-2)

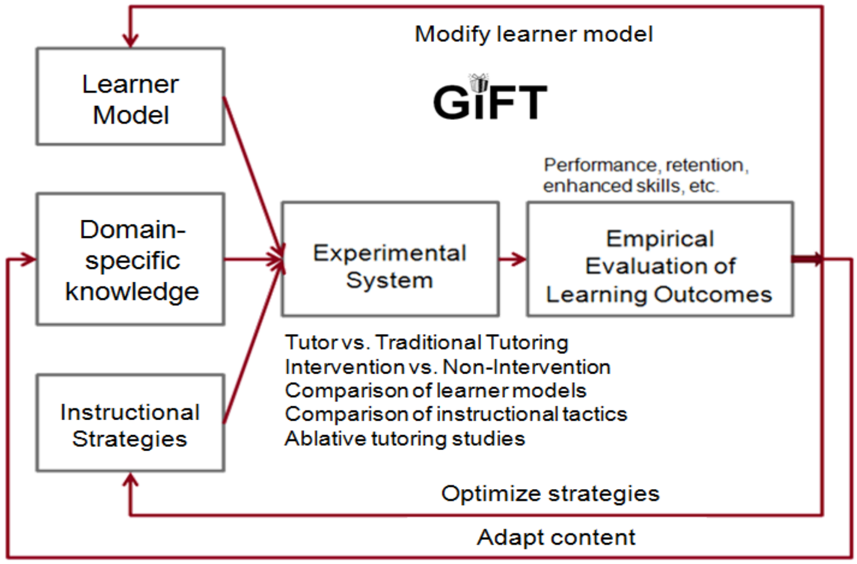

Figure P-2. GIFT Analysis Testbed Methodology

Figure P-2 illustrates an analysis testbed methodology being implemented in GIFT. This methodology was derived from Hanks, Pollack, and Cohen (1993) to allow manipulation of the learner model, instructional strategies, and domain-specific knowledge within GIFT, and support analysis of artificially-intelligent agents that influence the adaptive tutoring learning effect chain. In developing their testbed methodology, Hanks et al. reviewed four testbed implementations (Tileworld, the Michigan Intelligent Coordination Experiment [MICE], the Phoenix testbed, and Truckworld) for evaluating the performance of artificially intelligent agents. Although agents have changed substantially in complexity during the past 20–25 years, the methods to evaluate their performance have remained markedly similar.

The authors designed the GIFT analysis testbed based upon Cohen's assertion (Hanks et al., 1993) that testbeds have three critical roles related to the three phases of research. During the exploratory phase, agent behaviors need to be observed and classified in broad categories. This can be performed in an experimental environment. During the confirmatory phase, the testbed is needed to allow more strict characterizations of agent behavior to test specific hypotheses and compare methodologies. Finally, in order to generalize results, measurement and replication of conditions must be possible. Similarly, the GIFT analysis methodology (Figure P-2) enables the comparison/contrast of ITS elements and assessment of their effect on learning outcomes (e.g., knowledge acquisition, skill acquisition, and retention).

How to Use This Book

This book is organized into four sections:

I. **Fundamentals of Learner Modeling**
II. **Current Learner Modeling Tools and Methods**
III. **Emerging Learner Modeling Concepts**
IV. **Future Learner Modeling Concepts**

The *Fundamentals of Learner Modeling* section provides an overview of learner modeling terms and concepts along with discussion topics, and a review of the learner modeling literature. The *Current Learner Modeling Tools and Methods* section reviews current learner modeling tools and methods and provides design recommendations for GIFT and adaptive ITSs. The *Emerging Learner Modeling Concepts* section analyzes emerging learner modeling concepts and discusses their potential impact on design recommendations for GIFT and adaptive ITS. Finally, the *Future Learner Modeling Concepts* section projects how ITSs might be applied in the future and provides design recommendations to realize innovative capabilities in GIFT and adaptive ITSs.

Chapter authors in each section were carefully selected for participation in this project based on their expertise in the field as ITS scientists, developers, and practitioners. *Design Recommendations for Intelligent Tutoring Systems: Learner Modeling (Volume I)* is intended to be a design resource as well as community research resource that can be of significant benefit as the following:

- ***An educational resource for developing ITS scientists:*** Section I provides a wealth of information about ITS concepts and design, and presents an in-depth review of the learner modeling literature.

- ***A roadmap for ITS research opportunities:*** Sections II, III, and IV present current, emerging, and future concepts about learner modeling. This sampling of authors' perspectives is based on hundreds of cumulative years of experience in the ITS research domain and identifies significant gaps in current and emerging ITS technology (tools and methods). Each of these gaps points to yet unanswered research questions.

- ***A roadmap to the development and application of GIFT****:* As noted previously, GIFT is an open-source, publically available ITS architecture that is intended to make it easy to author ITSs; reduce the cost of ITS development by promoting reuse; automatically manage instruction based on best pedagogical practices; and allow scientists to compare and contrast evolving ITS capabilities to determine future best practices. As this book outlines issues and challenges associated with learner modeling, it also provides guidelines on how GIFT might be designed to address identified capability gaps. Future volumes of the *"Design Recommendations for Intelligent Tutoring Systems"* book series will provide insight to other ITS design domains including instructional strategy and tactics design, authoring and expert modeling, domain modeling, learning effect assessment, and team tutoring design. We encourage readers to become members of the GIFT community to build on its existing capabilities and support its future capabilities with us. More information on GIFT can be found by registering at www.GIFTtutoring.org. Registration provides access to GIFT source code, documentation, and related publications.

References

Aleven, V., McLaren, B., Roll, I. & Koedinger, K. (2006). Toward meta-cognitive tutoring: A model of help seeking with a cognitive tutor. *International Journal of Artificial Intelligence in Education, 16*, 101-128.

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

Anderson, L. W. & Krathwohl, D. R. (Eds.). (2001). *A taxonomy for learning, teaching and assessing: A revision of Bloom's Taxonomy of Educational Objectives: Complete edition*. New York : Longman.

Baker, R.S., D'Mello, S.K., Rodrigo, M.T. & Graesser, A.C. (2010). Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive-affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies, 68*, 223-241.

Cai, Z., Graesser, A.C., Forsyth, C., Burkett, C., Millis, K., Wallace, P., Halpern, D. & Butler, H. (2011). Trialog in ARIES: User input assessment in an intelligent tutoring system. In W. Chen & S. Li (Eds.*), Proceedings of the 3rd IEEE International Conference on Intelligent Computing and Intelligent Systems* (pp.429-433). Guangzhou: IEEE Press.

D'Mello, S. & Graesser, A.C. (2010). Multimodal semi-automated affect detection from conversational cues, gross body language, and facial features. *User Modeling and User-adapted Interaction, 20*, 147-187.

D'Mello, S. K., Graesser, A. C. & King, B. (2010). Toward spoken human-computer tutorial dialogues. *Human Computer Interaction, 25*, 289-323.

Doignon, J.P. & Falmagne, J. C. (1999). *Knowledge Spaces*. Berlin, Germany: Springer.

Elson-Cook, M. (1993). Student modeling in intelligent tutoring systems. *Artificial Intelligence Review, 7*, 227-240.

Gagne, R. M. (1985). *The conditions of learning and theory of instruction* (4th ed.). New York: Holt, Rinehart & Winston.

Goldberg, B.S., Sottilare, R.A., Brawner, K.W. & Holden, H.K. (2011). Predicting Learner Engagement during Well-Defined and Ill-Defined Computer-Based Intercultural Interactions. In S. D'Mello, A. Graesser, , B. Schuller & J.-C. Martin (Eds.), *Proceedings of the 4th International Conference on Affective Computing and Intelligent Interaction (ACII 2011) (Part 1: LNCS 6974)* (pp. 538-547). Berlin Heidelberg: Springer.

Graesser, A.C., Conley, M. & Olney, A. (2012). Intelligent tutoring systems. In K.R. Harris, S. Graham & T. Urdan (Eds.), *APA Educational Psychology Handbook: Vol. 3. Applications to Learning and Teaching* (pp. 451-473). Washington, DC: American Psychological Association.

Graesser, A. C., D'Mello, S. K., Hu. X., Cai, Z., Olney, A. & Morgan, B. (2012). AutoTutor. In P. McCarthy & C. Boonthum-Denecke (Eds.), *Applied natural language processing: Identification, investigation, and resolution* (pp. 169-187). Hershey, PA: IGI Global.

Graesser, A. C., Penumatsa, P., Ventura, M., Cai, Z. & Hu, X. (2007). Using LSA in AutoTutor: Learning through mixed initiative dialogue in natural language. In T. Landauer, D. McNamara, S. Dennis & W. Kintsch (Eds.), *Handbook of latent semantic analysis* (pp. 243–262). Mahwah, NJ: Erlbaum.

Graesser, A. C. & Person, N. K. (1994). Question asking during tutoring. *American Educational Research Journal*, *31*, 104–137.

Hanks, S., Pollack, M.E. & Cohen, P.R. (1993). Benchmarks, Test Beds, Controlled Experimentation, and the Design of Agent Architectures. *AI Magazine*, *14* (4), 17-42.

Heffernan,N.T., Koedinger, K.R. & Razzaq, L.(2008) Expanding the model-tracing architecture: A 3rd generation intelligent tutor for Algebra symbolization. *The International Journal of Artificial Intelligence in Education, 18*(2). 153-178.

Hu, X., Craig, S. D., Bargagliotti A. E., Graesser, A. C., Okwumabua, T., Anderson, C., Cheney, K. R. & Sterbinsky, A. (2012). The effects of a traditional and technology-based after-school program on 6th grade students' mathematics skills. *Journal of Computers in Mathematics and Science Teaching, 31*, 17-38.

Johnson, L. W. & Valente, A. (2008). Tactical language and culture training systems: Using artificial intelligence to teach foreign languages and cultures. In M. Goker & K. Haigh (Eds.), Proceedings of the Twentieth Conference on Innovative Applications of Artificial Intelligence (pp. 1632-1639). Menlo Park, CA: AAAI Press.

Krathwohl, D.R., Bloom, B.S. & Masia, B.B. (1964). *Taxonomy of Educational Objectives: Handbook II: Affective Domain*. New York: David McKay Co.

Lepper, M. R., Drake, M. & O'Donnell-Johnson, T. M. (1997). Scaffolding techniques of expert human tutors. In K. Hogan & M. Pressley (Eds), *Scaffolding learner learning: Instructional approaches and issues* (pp. 108-144). New York: Brookline Books.

Litman, D. (2013). Speech and language processing for adaptive training. In P. Durlach & A. Lesgold (Eds.), *Adaptive technologies for training and education*. Cambridge, MA: Cambridge University Press.

Mitrovic, A., Martin, B. & Suraweera, P. (2007). Intelligent tutors for all: The constraint-based approach. *IEEE Intelligent Systems, 22*(4), 38-45.

Murray, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education*, *10*(1), 98–129.

Murray, T. (2003). An Overview of Intelligent Tutoring System Authoring Tools: Updated analysis of the state of the art. In Murray, T.; Blessing, S.; Ainsworth, S. (Eds.), *Authoring tools for advanced technology learning environments* (pp. 491-545). Berlin: Springer..

Nkambou, R., Mizoguchi, R. & Bourdeau, J. (2010). *Advances in intelligent tutoring systems*. Heidelberg: Springer.

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

O'Neil, H. F. & Perez, R. (Eds.). (2003). *Technology applications in education: A learning view*. Hillsdale, NJ.: Erlbaum.

Patil, A. S. & Abraham, A. (2010). Intelligent and Interactive Web-Based Tutoring System in Engineering Education: Reviews, Perspectives and Development. In F. Xhafa, S. Caballe, A. Abraham, T. Daradoumis & A. Juan Perez (Eds.), *Computational Intelligence for Technology Enhanced Learning*. *Studies in Computational Intelligence* (Vol 273, pp. 79-97). Berlin: Springer-Verlag.

Person, N. K., Kreuz, R. J., Zwaan, R. A. & Graesser, A. C. (1995). Pragmatics and pedagogy: Conversational rules and politeness strategies may inhibit effective tutoring. *Cognition and Instruction, 13*(2), 161–188.

Picard, R. (2006). Building an Affective Learning Companion. Keynote address at the 8th International Conference on Intelligent Tutoring Systems, Jhongli, Taiwan. Retrieved from http://www.its2006.org/ITS_keynote/ITS2006_01.pdf

Psotka, J. & Mutter, S.A. (1988). *Intelligent Tutoring Systems: Lessons Learned*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Ritter, S., Anderson, J. R., Koedinger, K. R. & Corbett, A. (2007) Cognitive Tutor: Applied research in mathematics education. *Psychonomic Bulletin & Review, 14*, 249-255.

Rus, V. & Graesser, A.C. (Eds.) (2009). The Question Generation Shared Task and Evaluation Challenge. Retrieved from http://www.questiongeneration.org/.

Rus, V., McCarthy, P.M., McNamara, D.S. & Graesser, A.C. (2009). Identification of sentence-to-sentence relations using a text entailer. *Research on Language and Computation, 7*, 209-229.

Simpson, E. (1972). The classification of educational objectives in the psychomotor domain: *The psychomotor domain*. Vol. 3. Washington, DC: Gryphon House.

Sleeman D. & J. S. Brown (Eds.) (1982). *Intelligent Tutoring Systems*. Orlando, Florida: Academic Press, Inc.

Soller, A. (2001). Supporting social interaction in an intelligent collaborative learning system. *International Journal of Artificial Intelligence in Education, 12*(1), 40-62.

Sottilare, R. & Gilbert, S. (2011). Considerations for tutoring, cognitive modeling, authoring and interaction design in serious games. *Authoring Simulation and Game-based Intelligent Tutoring workshop at the Artificial Intelligence in Education Conference (AIED) 2011*, Auckland, New Zealand, June 2011.

Sottilare, R., Holden, H., Brawner, K. & Goldberg, B. (2011). Challenges and Emerging Concepts in the Development of Adaptive, Computer-based Tutoring Systems for Team Training. *Interservice/Industry Training Systems & Education Conference*, Orlando, Florida, December 2011.

Sottilare, R.A., Brawner, K.W., Goldberg, B.S. & Holden, H.K. (2012). *The Generalized Intelligent Framework for Tutoring (GIFT)*. Orlando, FL: U.S. Army Research Laboratory – Human Research & Engineering Directorate (ARL-HRED).

Sottilare, R. (2012). Considerations in the development of an ontology for a Generalized Intelligent Framework for Tutoring. *International Defense & Homeland Security Simulation Workshop* in Proceedings of the I3M Conference. Vienna, Austria, September 2012.

VanLehn, K. (2006) The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education*. *16*(3), 227-265.

VanLehn, K., Graesser, A. C., Jackson, G. T., Jordan, P., Olney, A. & Rose, C. P. (2007). When are tutorial dialogues more effective than reading? *Cognitive Science, 31*, 3-62.

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

Woolf, B.P. (2009). *Building intelligent interactive tutors*. Burlington, MA: Morgan Kaufmann Publishers.