Chapter 4

Intelligent Tutoring Systems, Serious Games, and the Generalized Intelligent Framework for Tutoring (GIFT)

Arthur C. Graesser, Xiangen Hu,

University of Memphis

Benjamin D. Nye, University of Southern California, and

Robert A. Sottilare

U.S. Army Research Laboratory

Many of us have had the vision of learners acquiring STEM subject matters by being immersed in motivating learning environments (such as games) that advance learners to new levels of mastery. Concepts in STEM (science, technology, engineering, and mathematics) are complex and difficult, and require learning at deeper levels than merely memorizing facts, rules, and procedures. Learners would ideally be challenged and motivated to improve on mastering complex topics that might not be acquired with traditional training methods. They would spend hundreds of hours in a hunt for a solution to a problem that few have solved, for the sweet spot in a trade-off between two or more factors, or for a resolution to a set of incompatible constraints. This is precisely the vision of progress for training in the 21st century. How can deep learning be achieved in a motivating learning environment? Games provide a good first place to look for answers because well-designed games are motivating and some meta-analyses have reported positive impacts of games on learning (Mayer, 2011; O'Neil & Perez, 2008; Ritterfeld, Cody, &

Vorderer, 2009; Shute & Ventura, 2013; Tobias & Fletcher, 2011; Wouters, van Nimwegen, van Oostendorp, & van der Spek, 2013).

This chapter explores the prospects of integrating games with intelligent tutoring systems (ITSs). The hope is that there can be learning environments that optimize both motivation through games and deep learning through ITS technologies. Deep learning refers to the acquisition of knowledge, skills, strategies, and reasoning processes at the higher levels of Bloom's (1956) taxonomy or the Knowledge-Learning-Instruction (KLI) framework (Koedinger, Corbett, & Perfetti, 2012), such as the application of knowledge to new cases, knowledge analysis and synthesis, problem solving, critical thinking, and other difficult cognitive processes. In contrast, shallow learning involves perceptual learning, memorization of explicit material, and mastery of simple rigid procedures. Shallow knowledge may be adequate for near transfer tests of knowledge/skills but not far transfer tests to new situations that have some modicum of complexity.

There have been some attempts to develop game-ITS hybrids (Adams & Clark, 2014; Halpern et al., 2012; Jackson & McNamara, 2013; Johnson & Valente, 2008; McNamara, Jackson, & Graesser, 2010; McQuiggan, Robison, & Lester, 2010; Millis et al., 2011; Sabourin, Rowe, Mott, & Lester, 2013). However, it is too early to know whether the marriage between games and ITSs will end up celebrating a multidecade anniversary or will end up in a divorce because of incompatible constraints between the two worlds. Deep learning takes effort, is often frustrating, and is normally regarded as work rather than play (Baker, D'Mello, Rodrigo, & Graesser, 2010; D'Mello, Lehman, Pekrun, & Graesser, 2014). Indeed, the correlation between liking and deep

learning tends to be negative in current ITS research without game attributes (Graesser & D'Mello, 2012; Jackson & Graesser, 2007). Perhaps game features can turn this work into play with sufficient entertaining features, learner freedom, and self-regulated activities (Lepper & Henderlong, 2000), and thereby shift the correlation from negative to positive (Sabourin et al., 2013). If not, then games may be reserved for the acquisition of shallow knowledge and skills, such as memorization of facts, simple skills, and rigid procedures. In contrast, games may not be suited for the acquisition of deep knowledge and strategies, such as understanding complex systems, reasoning about causal mental models, and applying sophisticated quantitative algorithms.

This chapter will not unveil the secrets of building a successful ITS in a game environment. It is too early to tell that story. Instead, we hope to achieve three goals. First, we will review successes and challenges in ITS research and development. Second, we will describe the components of ITSs in the Generalized Intelligent Framework for Tutoring (GIFT). GIFT has recently been developed by the U.S. Army Research Laboratory as a stable blueprint and guide for developing ITSs in the future (Sottilare, Graesser, Hu, & Holden, 2013; Sottilare, Graesser, Hu, & Goldberg, 2014). Third, we will reflect on how these efforts might be integrated with games. An adequate understanding of ITS components and the underlying research is a necessary prerequisite to formulating a meaningful courtship between ITS and games.

It is important to point out two areas of research and application that will not be addressed in this chapter. This chapter does not address the role of games in the acquisition and mastery of shallow learning. The empirical evidence has convinced us that a well-designed game can

effectively enhance shallow learning, whereas there is uncertainty in the literature on whether deep learning can benefit from games. This chapter also does not address learning and problem solving in the context of teams. Our focus is on deep learning in individuals who interact with an ITS.

Successes and Challenges in ITS Research and Development

This section briefly defines what we mean by an ITS, reviews the successes of ITS technologies, and identifies the chief challenges in scaling up these systems for more widespread use. Meta-analyses and landmark systems support the claim that ITSs are a promising solution to achieving deep learning. However, there are four categories of challenges which we place under the umbrellas of motivation, measurement, materials, and money. These challenges can be mitigated, if not conquered, by some recommended efforts.

What Is an Intelligent Tutoring System?

We define an ITS as a computer learning environment that helps the student master deep knowledge/skills by implementing powerful intelligent algorithms that adapt to the learner at a fine-grained level and that instantiate complex principles of learning (Graesser, Conley, & Olney, 2012). We see ITS environments as a generation beyond conventional computer-based training (CBT). CBT systems also adapt to individual learners, but they do so at a more coarse-grained level with simple learning principles. In a prototypical CBT system, the learner (a) studies material presented in a lesson, (b) gets tested with a multiple-choice test or another objective test, (c) gets feedback on the test performance, (d) re-studies the material if the performance in c is below threshold, and (e) progresses to a new topic if performance exceeds

threshold. The order of topics presented and tested typically follows a predetermined order, such as ordering on complexity (simple to complex) or ordering on prerequisites. The materials in a lesson can vary from organized text with figures, tables, diagrams, and multimedia to example problems to be solved. ITSs can be viewed as enhancements of CBT with respect to the adaptability, grain-size, and the power of computerized learning environments. In ITS, the processes of tracking knowledge (called user modeling) and adaptively responding to the learner incorporate computational models in artificial intelligence and cognitive science, such as production systems, case-based reasoning, Bayes networks, theorem proving, and constraint satisfaction algorithms (see Graesser, Conley, & Olney, 2012; Woolf, 2009).

This chapter does not sharply divide systems that are CBT systems versus ITSs, but one useful dimension is the space of possible computer-learner interactions that can be achieved with the two classes of systems. For an ITS, every tutorial interaction is unique and the space of possible interactions is extremely large, if not infinite. Imagine hundreds of alternative states of the learner, hundreds of alternative responses of the tutor, and thousands/millions of alternative sequences of interaction. An ITS attempts to fill in very specific learning deficits, to correct very specific misconceptions, and to implement dynamic sequencing and navigation. For CBT, interaction histories can be identical for multiple students and the interaction space is finite, if not small (e.g., < 100 possible interactions).

Successful ITSs have been developed for mathematically well-formed topics, including algebra, geometry, programming languages (the Cognitive Tutors: Aleven, McClaren, Sewall, & Koedinger, 2009; Anderson, Corbett, Koedinger, & Pelletier, 1995; Koedinger, Anderson,

Hadley, & Mark, 1997; Ritter, Anderson, Koedinger, & Corbett, 2007; ALEKS: Doignon & Falmagne, 1999), physics (Andes, Atlas, and Why/Atlas: VanLehn et al., 2002; VanLehn et al., 2007), electronics (SHERLOCK: Lesgold, Lajoie, Bunzo, & Eggan, 1992), and information technology (KERMIT: Mitrovic, Martin, & Suraweera, 2007). Some intelligent systems handle knowledge domains that have a stronger verbal foundation as opposed to mathematics and precise analytical reasoning. AutoTutor (Graesser, Chipman, Haynes, & Olney, 2005; Graesser, D'Mello, et al., 2012; Graesser et al., 2004; Nye, Graesser, & Hu, 2014) helps college students learn about computer literacy, physics, and critical thinking skills by holding conversations in natural language. Other natural language ITSs that have shown learning gains include DeepTutor (Rus, D'Mello, Graesser, & Hu, 2013), iSTART (McNamara et al., 2010), and My Science Tutor (Ward et al., 2013). The Intelligent Essay Assessor (Landauer, Laham, & Foltz, 2003) and e-Rater (Burstein, 2003) grade essays on science, history, and other topics as reliably as experts of English composition. These systems automatically analyze language and discourse by incorporating recent advances in computational linguistics (Jurafsky & Martin, 2008; McCarthy & Boonthum-Denecke, 2012) and information retrieval, notably latent semantic analysis (Landauer, McNamara, Dennis, & Kintsch, 2007).

Meta-analyses

Meta-analyses and reviews support the claim that ITS technologies routinely improve learning over classroom teaching, reading texts, and/or other traditional learning methods. These meta-analyses normally report effect sizes (sigma, σ), which refer to the difference between the ITS condition and a control condition in standard deviation units. The reported meta-analyses show positive effect sizes that vary from $\sigma = 0.05$ (Dynarsky et al., 2007) to $\sigma = 1.08$ (Dodds &

Fletcher, 2004), but most hover between $\sigma = 0.40$ and $\sigma = 0.80$ (Ma, Adesope, & Nisbett, in press; Fletcher, 2003; Graesser, Conley, & Olney, 2012; Steenbergen-Hu & Cooper, 2013, 2014; VanLehn, 2011). Our current best meta-meta estimate from all of these meta-analyses is $\sigma = 0.60$. This performance is comparable to human tutoring which varies from between $\sigma = 0.20$ and $\sigma = 1.00$ (Cohen, Kulik & Kulik, 1982; Graesser, D'Mello, & Cade, 2011), depending on the expertise of the tutor. Human tutors have not varied greatly from ITSs in direct comparisons between ITSs and trained human tutors (Olney et al., 2012; VanLehn, 2011; VanLehn et al., 2007).

We are convinced that some subject matters will show higher effect sizes than others when comparing any intervention (e.g., computer trainers, human tutors, group learning) to a control. It is difficult to obtain high effect sizes for literacy and numeracy because these skills are ubiquitous in everyday life and habits are automatized. For example, Ritter et al. (2007) reported that the Cognitive Tutor for mathematics has shown an effect size of $\sigma = 0.30$ to 0.40 in environments with minimal control over instructors. Human interventions to improve basic reading skills typically report an effect size of $\sigma = 0.20$. In contrast, when the student starts essentially from ground zero, such as many subject matters in science and technology, then effect sizes are expected to be more robust. ITSs show effect sizes of $\sigma = 0.60$ to 2.00 in the subject matters of physics (Van-Lehn, 2011; VanLehn et al., 2007), computer literacy (Graesser et al., 2004; Graesser, D'Mello, et al., 2012), biology (Olney et al., 2012), and scientific reasoning (Millis et al., 2011; Halpern et al., 2012). As a notable example, the Digital Tutor (Fletcher & Morrison, 2012) improves information technology by an effect size as high as $\sigma = 3.70$ for

knowledge and $\sigma = 1.10$ for skills. Such large effect sizes would never be expected in basic literacy and numeracy.

Motivation

ITS technologies that target deep learning have the challenge of keeping students motivated because, as mentioned earlier, the intrinsic tendency is for there to be an inverse relationship between liking a system and deep learning. Simply put, thinking and reasoning hurt. The hope is that games will fill in the motivational gap for ITSs. Unfortunately, there have not been enough studies that combine games with ITSs for a meta-analysis at this point in history. However, some example successful game-ITS hybrids have been Crystal Island (McQuiggan et al., 2010; Sabourin et al., 2013), iSTART-ME (Jackson, Dempsey, & McNamara, in press), and Operation ARIES and Operation ARA (Halpern et al., 2012; Millis et al., 2011). However, there is not an adequate body of research that reports effect sizes that contrast the game versions versus those without game features in these ITSs.

Game elements take some time to master so there is the risk of short-term penalties from games (Adams, Mayer, McNamara, Koenig, & Wainess, 2012). Narrative, fantasy, competition, choice, feedback, challenge, and other distinctive characteristics of games (Ritterfeld et al., 2009) may help motivation but they are not often intrinsic to subject matter mastery. In essence, game elements may pose a non-germane load to working memory and be a distraction from deep learning. This problem could be circumvented if all game elements had tangible hooks to the subject matter, but it is very rare to have game affordances aligned with components of deep learning. Given this difficulty of mapping game features to serious subject matter, the central

question is whether the game features will have a payoff in the long run. For example, the game features of iSTART-ME (Jackson et al., in press) had a short-term penalty compared to an ITS without game features, but iSTART-ME showed advantages after 8-10 hours. Therefore, we would argue that an adequate test of game characteristics for deep learning should involve assessments for 10 or more hours. Short interventions of an hour or less are essentially irrelevant because deep learning by definition takes many hours of training until mastery.

An ideal assessment of the motivational influence of game features on an ITS would allow students a free choice on whether to interact with the system. In essence, there would be a race horse comparison in the total amount of learning when the ITS does versus does not have the game features. A mathematical integral metric is needed to incorporate both time and learning (much like integral calculus). An hour of training might show the game version to be only .7 as effective as the standard version without the game, but what if the learner chooses to play the game version 10 times as long as the standard version? That would be a substantial long-term victory for the game version. There needs to be a learning gain metric that multiplies *learning efficiency* (e.g., learning-per-hour) times *time* (number-of-hours) that the learner voluntarily uses the learning environment in a free-choice or self-regulated learning scenario after a learning environment is exposed to the learner. However, such an integral metric is virtually never reported because studies attempt to equilibrate time on task between conditions. These studies ignore or diminish the motivational dimension of the Learning × Motivation equation.

Measurement

All learning systems need to be assessed by defensible measures of performance and learning. The validity and reliability of measures are a broad and important matter that is ubiquitously discussed among researchers that range from laboratory scientists to stakeholders of international assessments. The standards vary among research communities, and this applies to those within the ITS field. Ideal metrics have not been identified within the ITS community, let alone those involved with high-stakes state, national, and international assessments. Since the goal is deep learning, there would ideally be a psychometrically validated metric of deep learning for each particular subject matter. Unfortunately, available psychometric measures are a mixture of deep and shallow learning, as well as relevant versus irrelevant knowledge/skills. This is because they are generic measures of a broad skill rather than a metric that targets the specific subject matter of the ITS. Therefore, there is rarely a defensible gold standard for assessment of an ITS. In the absence of a suitable psychometric measure, researchers turn to researcher-defined metrics. Unfortunately, there is a risk of tailoring the ITS to the test under these circumstances, which makes it difficult to compare performance across studies and ITS technologies.

Another measurement problem lies in identifying the correct performance parameter. The typical metric is a learning gain metric that compares performance in a posttest with a pretest; there is either a difference score [post-pre] or a statistical analysis of posttest scores that partials out contributions of pretest scores. The learning gains are compared for the ITS versus the comparison condition. Sometimes normalized learning gains are computed that adjust for the pretest level: [(post-pre)/(1 – pre)]. Researchers also occasionally collect learning gain data for specific principles and concepts (Forsyth et al., 2012; VanLehn et al., 2007) and average these

gains for the total set of principles/concepts. Arguments are also made for collecting learning parameters, namely how fast the learning occurs.

One parameter that may be considered is learning efficiency, which computes the amount of learning per unit of study time, that is, learning gain per hour. We argue that such a metric is inappropriate for any ITS that is targeting deep learning and mastery of the subject matter, unless it is computed appropriately. There are two problems with any simple metric of learning efficiency. First, it does not guarantee mastery of the deep knowledge/skill. Learning environments suited for shallow learning may plateau for deep knowledge and never meet the threshold of mastery, even after hundreds of hours of training. Second, metrics often include a combination of shallow and deep knowledge/skills. When that occurs, the efficiency metric is excellent during the initial time window by virtue of shallow learning, but it never reaches the threshold of mastery for deep learning. An appropriate performance measure for an ITS that targets deep learning should include exclusively deep performance indicators. Of course, a separate performance measure could be computed for shallow learning, but it is the worry of acquiring the deep knowledge/skills that is the central bottleneck.

We argue that an appropriate learning metric for an ITS would satisfy a number of criteria. First, the researchers need to decide on a set of knowledge/skills to master and a threshold for each that specifies adequate mastery. Second, the researchers need to measure the amount of training time (and/or the rate of learning) until mastery is reached for each knowledge/skill. Third, the researchers need to measure the total training time to master the total set of knowledge/skills. Our conjecture is that a good ITS will eventually meet these criteria whereas conventional

trainers will either fail to reach performance thresholds or will take much more time to reach mastery of deep knowledge/skills.

Materials

The developers of the materials in most of today's ITSs require at least three forms of expertise: Subject matter knowledge, computer science, and pedagogical strategies. Subject matter expertise will always be needed, but there has been the dream of creating authoring tools that minimize expertise in computer science (Ainsworth & Grimshaw, 2004; Aleven et al., 2009; Murray, Blessing, & Ainsworth, 2003). The authoring tools would be so easy to navigate and use that only modest expertise in information technologies would be adequate for a subject matter expert to create the learning materials. Imagine teachers in K12 and designers of MOOCs (Massively Open Online Courses) being able to develop materials for ITS environments. Imagine Nobel Laureates creating materials in less than a month to be shared directly with the world through ITS technologies. Unfortunately, the complexity of current ITS technologies has been a major challenge to this lofty goal. With rare exception, those who create the materials for current ITSs have moderate to high computer science expertise. Moreover, their knowledge of pedagogy is unspectacular.

We argue that the authoring tool bottleneck is best confronted by developing a science of authoring processes that helps the field better understand what training and information resources are needed. Just as there are sciences of writing, engineering design, software development, and other creative endeavours, there needs to be a science of creating materials for advanced learning

environments in the future. Without a systematic science of the cognitive processes, technologies, and metrics of assessment, the bottleneck will continue to exist.

Money

The expense of developing an ITS is often expressed as a concern of those who make budget decisions (Fletcher, 2014; Fletcher & Morrison, 2012). Graesser and King (2008) projected the following estimates of costs: "Approximate costs for an hour training session with conventional computer-based training would be $10,000, for a 10-hour course with conventional computer-based training and rudimentary multimedia would be $100,000, for an information-rich hypertext-hypermedia system would be $1,000,000, for a sophisticated intelligent tutoring system would be $10,000,000, and for a serious game on the web with thousands of users would be $100,000,000" (pp. 130). Colleagues have raised questions and have sometimes disagreed with these estimates, but we would argue that the estimates are within an order of magnitude of being correct.

The internet entirely changes the landscape on costs. A learning environment that costs $100 million to develop is inexpensive if it can be delivered to 10 million people, but too expensive if only to 10 people. The population of course delivery is therefore very important in the consideration of costs. We would argue that it is also very important to consider the depth of the knowledge/skills. Higher cost is essential if it is the only way for the students to receive deep knowledge/skills. A $1 million system is worthless if it never progresses students beyond shallow knowledge and if depth is required. There need to be concrete answers to stakeholders

on the costs for achieving the targeted levels of expertise in addition to planning, developing, testing, and scaling up any ITS.

A number of concrete answers have been identified to lower costs and meet the pedagogical requirements of deep learning. Four solutions are addressed here. First, there needs to be standards for reusing learning objects in different systems in a manner that supports smooth interoperability between systems. The military took the lead with their Advanced Distributed Learning initiative (Fletcher, 2009) and the development of SCORM (Shareable Content Object Reference Model) standards for learning objects. Suppose that a chestnut learning object of 1-10 minutes is developed by any creative instructor in the internet universe. If it has the right structure and metadata, it can be shared with millions of others and incorporated in an ITS. It takes only one chestnut learning object to meet standards and once that is achieved it can go viral and save costs. Second, authoring tools can be used to develop new learning objects with the ideal content, constraints, and metadata to be shared with other learning management systems. The authoring tools will of course need to be designed to maximize interface design for those with minimal computer science experience. Third, there needs to be a computational infrastructure to support these goals of sharability, interoperability, reuse, and so on. And fourth, it is important to tap into the successful ITS technologies that have already been built. There have been three decades of ITS development for basic universal skills, such as mathematics, physics, engineering, reading, and scientific reasoning. We need to capitalize on these landmark investments.

Generalized Intelligent Framework for Tutoring (GIFT)

The Generalized Intelligent Framework for Tutoring (GIFT) architecture is a major initiative by the Army Research Laboratory that targets some of these core ITS roadblocks (Sottilare et al., , 2013; Sottilare et al., 2014). From the standpoint of the present book, the hope is that a systematic architecture (such as GIFT) will help overcome obstacles in building serious games in a manner that minimizes costs and development time, but maximizes student learning and motivation. Two roadblocks that GIFT concentrates on are the lack of modularity and the lack of shared standards, in addition to the other challenges articulated in this previous section. GIFT has three high-level components that are widely acknowledged in computer-based learning communities:

1. Standards-based, modular ITS components (i.e., learner models, pedagogy modules) and authoring tools to support authoring for these components,
2. An instructional manager that facilitates selection from the best pedagogical strategies, and
3. A testbed to study the impact of different ITS components and pedagogical strategies on learning.

This section is pitched at a somewhat technical level that can accommodate a diverse set of learner models and pedagogical strategies. An architecture such as GIFT guides curriculum designers, empirical researchers, and software developers in a coordinated manner. GIFT has multiple complementary functionalities: a service specification for connecting ITS components, the specific ITS components implemented by the standard GIFT runtime, and authoring tool suites. GIFT fulfills these objectives by adhering to modular design principles. That is, it needs to separate components so that they can be substituted for others that perform similar functions.

These principles are also important for integration with third-party systems (e.g., game worlds), as they impose a well-defined interface for communicating with new systems.

GIFT addresses a serious challenge for the ITS community that has been recognized for over a decade. As discussed earlier, a major blocking point for scaling up ITSs has historically been the cost of development. This is particularly important for game-based ITSs that must incorporate complex tutoring functionality into an often already complex gaming environment. One solution is to tightly integrate the game environment with tutoring. Well-established systems such as Crystal Island (Rowe, Shores, Mott & Lester, 2011) and Operation ARIES and Operation ARA (Halpern et al., 2012; Millis et al., 2011) use tight integration, such that several ITS principles and algorithms impose significant constraints over the entire game. The good news is that both of these systems have shown learning gains at deeper levels. One potential liability of these complex systems is that they scale poorly whenever a custom solution is required for each new game, unless the system can be decomposed into functional pieces.

The GIFT architecture takes an approach that emphasizes a loose, service-based integration of tutoring systems into games (Sottilare, Goldberg, Brawner, & Holden, 2012). Thus, the game environment imposes most of the constraints, whereas ITS principles are woven into the game to enhance the game. To date this has been accomplished technically with two serious games (Nye, Hu, Graesser, & Cai, in press): Virtual BattleSpace 2 (a first-person shooter game) and VMedic (a combat casualty care game, Engineering and Computing Simulation, 2012). There are advantages to having ITS components being modular to the point of being used in many different games. It increases reuse of components, which has strong practical benefits. Direct development

costs for transferring tutoring to new game platforms are reduced, since only platform-specific mechanisms need to be redesigned, rather than the entire pedagogical decision-making system. Unfortunately, however, empirical data are not available that assess whether Virtual BattleSpace 2 and VMedic help student learning or motivation, and also whether there are major reductions in development time and costs. Such assessments are currently underway.

It is important to emphasize that GIFT is designed to increase quality but simultaneously decrease development costs. Tutoring can be developed for one game, then ported to a second game with similar content. This is a general benefit of modularity and separation of components in software design: Building components and strategies that are highly portable allows researchers to design components and then have them tested and refined more effectively. The researchers who use the tools may vary in expertise, ranging from computer scientists to curriculum developers who have limited computer technology skills. Modularity also allows GIFT to use the same suite of authoring tools across multiple domains and learning environments. GIFT is a relatively new architecture, so the magnitude of such benefits remains unclear. We argue that the ability to build or modify an ITS "piece by piece" is an important avenue that could drastically reduce barriers to developing ITS in the long term.

GIFT Real-Time Adaptive Components

The major GIFT real-time components are summarized in Nye, Sottilare, Ragusa, & Hoffman, 2014, see Figure 4.1 of that article). There is a Tutor-User Interface that interprets the input of the learner and transmits system actions to the learner environment. A Gateway Module acts as a bridge to third-party environments, ranging from 3D gaming environments to productivity

applications such as Microsoft PowerPoint. The Gateway module allows the rest of the system to remain separate from the specific game or learning environment. Multiple modules may split this functionality, such as SIMILE (Student Information Models for Intelligent Learning Environments), a dedicated system for monitoring performance in a learning environment (Engineering and Computing Simulations, 2012). A Sensor Module acts as an interface to third-party sensors, such as biofeedback sensors and emotion classifiers. Such components are increasingly popular in ITS research as researchers explore the roles of motivation and affect in learning (Calvo & D'Mello, 2010; D'Mello & Graesser, 2010, 2012; McQuiggan et al., 2010). A Domain Module manages information about the specific domain of instruction (e.g., algebra, military medicine, etc.), which is read from a Domain Knowledge File (DKF) for the current tutoring domain. A Learner Module tracks learners' knowledge, performance, emotion, and social states and thereby determines how well they have mastered the material and estimates their capabilities for future interactions. Additional learner information may be communicated to the learner module by external systems, such as the Learning Record Stores (LRS) that maintain biographical data and historical learning data. Finally, a Pedagogical Module contains instructional strategies that can be selected during a session. These strategies determine the strategies and skills that guide how GIFT intervenes to improve learning.

GIFT intervenes in a gaming environment by monitoring the states and shifts in the learner's state, and then using these shifts to select instructional strategies. The goals of GIFT strategies are intended to increase domain knowledge, but in a game environment there are also the goals of maintaining motivation and persistence. The representation of pedagogical strategies in GIFT consists of IF <state> THEN <action> production rules, a standard representation for

strategically selecting instructional strategies. Rule-based tutoring strategies have a long history in ITS (Anderson et al., 1995; Graesser, Conley, & Olney, 2012; Woolf, 2009). The system watches over the landscape of current states existing in the working memory. Then, if particular states exist or reach some threshold of activation, then a production rule is fired probabilistically. Contemporary rules are never brittle, but rather are activated to some degree and probabilistically. GIFT strategies are intended to be domain independent and are later resolved via domain dependent tactics that are specific to the instructional domain and that activate actions in the game environment (Nye, Sottilare, Ragusa, Hoffman, 2014).

The general GIFT processes and components provide multiple levels of adaptivity to learners. The example in Figure 4.1 focuses on microadaptive behavior, also sometimes referred to as the inner-loop or step-based tutoring (VanLehn, 2006). Microadaptivity occurs when a system supports the user on one or more ongoing tasks or goals. GIFT can also provide macroadaptive support for learning, sometimes called outer-loop adaptivity. Macroadaptivity includes selecting tasks or problems for the learner to solve, usually with the intention to keep problems within a learner's zone of proximal development.

GIFT Information Flow

A simplified view of GIFT information flow can be considered when providing real-time microadaptation. GIFT is under active development, so some of these functions are likely to evolve over time (the current version is GIFT 4.0). Rather than focusing on specific details or mechanisms, we trace how knowledge flows through the system in order to explore how data about the learner, the learning environment, and the domain are processed by instructional

strategies in GIFT to produce meaningful pedagogical actions for a gaming environment. Table 4.1 gives a high-level overview of how GIFT uses strategies and tactics to select actions that impact the learner as they interact with a game. Each of these steps will be described briefly. Symbols are assigned to various information states and functions noted in Table 4.1 to facilitate referencing the information in each step.

< TABLE 4.1 HERE>

Session Inputs: Sensors and the Learning Environment (Step 1)

GIFT strategies have three main sources of information: the learning environment for the user (1.A), the external sensor data streams (1.B), and the model of the learner based on accumulated events over time (1.C). The learning environment could include the user interface, the state of the game world, or possibly the state of an accompanying slideshow presentation. Events with information about user behavior (**E_t**) are sent in real time to GIFT from the communication module for the learning environment. External sensors may also provide information about the learner (**D_t**), such as biometrics and emotion classifiers. Each of these input sources reaches the learner module by a separate path. While learner behavior from the communication module passes through the domain module (Steps 2-4), sensor information is directly fed to the learner module. Many sources of information are integrated into the learner module (**D_L**), such as persistent learner data (i.e., data stored in a learning management system) or biographical data (e.g., gender, age, etc.). Considerable information is not likely to change during a single learning session, so they will be treated as invariant for this discussion and will reside in 1.C. However, in practice, their states may change during a session.

Assessing Performance: Domain Module (Steps 2-4)

The domain module uses the events and information from the learning environment to assess performance on a set of domain concepts (**C**). Two types of concepts exist: low-level concepts ($\mathbf{C_L}$) that are evaluated based on performance assessment rules and higher level concepts ($\mathbf{C_H}$) where performance is inferred from performance on lower level ones. Thus, there is a hierarchical structure of grain size, with a threshold differentiating low from high. Assessment rules are stored in a domain-knowledge file (DKF), which contains all the domain-specific rules and concepts for the tutoring system. For each low-level concept, the performance assessment rules for each concept ($\mathbf{R_c}$) use the learning environment events to classify performance as "Below Expectation," "At Expectation," "Above Expectation," or "Unknown" (for when it is not yet assessed). For higher level concepts, performance is derived from the performance of child concepts through an aggregation function that "rolls up" performance ($\mathbf{F_c}$). External performance assessments can also be received, such as those calculated by SIMILE, a dedicated system for monitoring performance in a learning environment (Engineering and Computing Simulations, 2012). Third-party systems such as SIMILE can calculate and transmit assessments from a game environment, acting as a bridge between GIFT and a specific gaming environment. After performance is assessed, these assessments are sent to the learner module and pedagogy module.

Strategy Selection: Learner and Pedagogy Modules (Steps 5-7)

Performance and sensor data are considered in either discrete or continuous states. A large enough change to any of these states in the learner module can trigger a search for an appropriate strategy to support the learning goals. This selection process is handled by the pedagogical module, which considers the current learner state (sensors and learning assessments) and the

prior learner state. For any given transition, one instructional strategy may be selected by the pedagogy module. The strategy selection process ($\mathbf{F_S}$) is determined by functions in the pedagogy module, which may be rules (similar to the DKF) or more advanced Java functions. In general, rules are used and different transitions may be combined using Boolean operators (e.g., AND) to determine the conditions for selecting a domain-independent strategy decision. For many transitions, no strategy may be activated. In that case, the pedagogical module waits until the next strategy trigger occurs. When a strategy ($\mathbf{S_t}$) is selected, it is sent to the Domain Module for evaluation. At this stage, the strategy decision may be referred to as an "abstract strategy" because it is not domain specific.

As a concrete example, consider the transition of a student being engaged in a task versus disengaged or bored. This is a transition that can be sensed from multiple channels with some degree of accuracy (Calvo & D'Mello, 2010; D'Mello & Graesser, 2010). One selected pedagogical strategy would be to increase or decrease the difficulty of the assigned next task (D'Mello & Graesser, 2012), which would depend on the performance level of the student in the session. For comparatively higher performers, more difficult tasks would be selected in order to increase the challenge level. For lower performers, easier tasks would be selected because the existing difficulty level is beyond what the student can handle. So the general abstract pedagogical strategy in this example is to adjust the difficulty level of the next task when there is a large discrepancy in engagement and the adjustment depends on specific knowledge states and performance of the learner. Production rules capture these contingencies in GIFT and there is empirical evidence for some of these production rules. An affect-sensitive AutoTutor has been shown to improve learning in comparison with an affect-neutral AutoTutor (D'Mello &

Graesser, 2012) but it is too early to quantify effect sizes for particular strategies at this point in the science.

From Strategies to Tactics: Domain Module (Steps 8-9)

After a general strategy is selected, it must be translated into a more specific form that is suitable for the domain. For example, an abstract strategy decision might be: "Provide corrective feedback for Concept A." The Tactics component of the domain module must map each strategy to a domain-specific decision. At present, tactics (**T**) are mapped on a one-to-one basis to abstract strategy decisions (**S**) in GIFT. This mapping ($\mathbf{F_A}$) is defined as part of the domain-specific information. As an example, a tactical decision for a math domain to "Provide corrective feedback for Concept A" would be to inform the learner that the correct answer is 5, but their answer was 8. Alternatively, for a medical domain, Concept A might be a diagnosis, so tactical design could inform the learner that their diagnosis of anemia was wrong and that the correct answer was scurvy. Once a tactical decision has been made, this decision ($\mathbf{T_t}$) will be sent to the learning environment, which will take some actions that will implement the decision. So continuing the example of corrective feedback, the learning environment might provide a voiceover that speaks the corrective feedback. In a different learning environment, this feedback might be provided using a text hint instead. This modularity makes GIFT well-matched for integrating intelligent tutoring into a variety of game environments.

Closing Comments

The technical specifications of GIFT help organize the ITS side of the GIFT-ITS marriage. That is, all ITSs must somehow fit into the GIFT conception of ITSs. However, what about the

motivating game elements? Our conjecture is that game components can fit in the same architecture with little or no problems other than understanding the essence of games. Game features are essentially like any other subject matter, namely complex, multifaceted, and ranging from brittle to probabilistic in is mechanism. Just as math can be merged with physics, so can games. Angry Birds and Newton's Playground (Shute & Ventura, 2013) are success cases in illustrating the meshing of game constraints with formal systems.

But alas, games have some constraints that are very different than the components of deep learning and that will pose challenges to meshing the worlds (Adams & Clark, 2014; Graesser, Chipman, Leeming, & Beidenbach, 2009). We believe it is most feasible to embed ITS modules within existing game environments to enhance the game, such as intelligent dialogue, simulations, and so on. The native motivational features of a successful game will be minimally compromised by the embedded intelligent features. We believe it is possible to add game features to ITS and thereby attempt to enhance motivation (called gamification), but that may have limited success for reasons articulated below. Finally, we believe that it will be extremely difficult to develop a game that has components that are closely aligned with the constraints of an ITS because the constraints are very different. Below are some of the pressure points that may make it difficult, or even impossible, to design some game-ITS technologies that show benefits for deep learning.

1. **Non-germane load bloat**. The cognitive load from the game elements may not be germane to the mastery of the serious subject matter and ultimately reduce deep learning (Adams et al., 2012). For example, if the narrative, fantasy, and competition

components take up too much time and are profoundly distracting, then an insufficient amount of deep learning may be achieved. The penalty may persist over and above the added time the game elements afford for intrinsic motivation and self-regulated learning. When this occurs, there are no payoffs for the game elements on any metric, including the integral learning-time metric discussed earlier.

2. **Feedback guideline clashes**. Feedback is an important aspect of both ITSs and games. However, the timing and nature of the feedback may be very different for the two worlds. Games often provide timely, if not quick, feedback to the learner about the quality of their contributions in order to keep the student in what Csikszentmihalyi (1990) called the state of psychological flow. Flow is intense engagement to the point where time and sometimes fatigue psychologically disappear. In ITS technologies, there needs to be time for thought and reflection over the depth of the material, a timing pattern that might clash with the speedy tempo of games. In essence, there will be a clash in timing if online temporal dynamics are incompatible in games and ITSs. There may also be traffic jams among feedback, particularly in complex environments with many competing tasks. Prioritizing feedback in a dynamic, game-based environment is non-trivial. Similarly, there may be clashes in the qualitative feedback, such as justifications, explanations, and recommended actions. Qualitative feedback is perhaps the hallmark of ITS deep learning, but hard core gamers may not appreciate technical content encroaching on their game experience. The serious content needs to be smuggled into games in slick ways that routinely stymie game designers that aspire to build serious games.

3. **Content collision between narrative and deep learning**. The ideal is a seamless harmony between the game narrative and the subject matter content. Unfortunately, the odds of that happening may be akin to a film director winning an Academy Award. What is the typical integration scenario? Either the narrative does not promote the difficulties of the subject matter, or the narrative is incoherently boring as it caters to the constraints of the subject matter. It is safe to assume that the two worlds are in collision unless a genius can find ways to connect them. That being said, there may be some realistic approaches in meshing narrative with ITSs to promote deep learning. Specifically, the ITS modules can be embedded within the game world to increase the intelligence of game components and to avoid interfering with the conceptual integrity of the game constraints.

4. **Control struggles**. The learners want to be in control and follow their whims in a capricious trajectory that is guided by intrinsic motivation or possibly self-regulated learning. The harbingers of deep knowledge want to be in control over the learning experience to satisfy the curriculum, pedagogy, and efficiency metrics. This is a power struggle.

5. **System engineering disconnections**. These various incompatible constraints might possibly be resolved by a cost-benefit analysis that maximizes progress. That will not happen if stakeholders wallow in their professional caves, guard their positions, and

resist communication and compromise. A cost-benefit analysis needs to be quantified in monetary units.

It is very true that there are struggles in solving anything fundamental to society. However, we continue to be skeptically optimistic on promoting the game-ITS marriage because the lofty goal of turning work into play may be in sight with enough effort, coordination, science, and creativity. We need to see more success cases of systems that apply game features to intelligent tutoring systems, that weave ITS modules into game components, and that have successful dances between the constraints of games and subject matter domains. More success cases are needed before we can answer the question of whether serious games can promote deep learning of difficult academic material.

References

Adams, D. M., & Clark, D. B. (2014). Integrating self-explanation functionality into a complex game environment: Keeping gaming in motion. *Computers & Education, 73,* 149-159.

Adams, D. M., Mayer, R. E., McNamara, A., Koenig, A., & Wainess, R. (2012). Narrative games for learning: Testing the discovery and narrative hypotheses. *Journal of Educational Psychology, 104*(1), 235-249.

Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K. (2009). A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education, 19,* 105-154.

Ainsworth, S. E., & Grimshaw, S. K. (2004). Evaluating the REDEEM authoring tool: Can teachers create effective learning environments? *International Journal of Artificial Intelligence in Education, 14,* 279-312.

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

Baker, R. S. J. d., 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.

Bloom, B. S. (1956). *Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive Domain.* New York: McKay.

Burstein, J. (2003). The E-rater scoring engine: Automated essay scoring with natural language processing. In M. D. Shermis & J. C. Burstein (Eds.), *Automated essay scoring: A cross-disciplinary perspective* (pp. 122-133). Mahwah, NJ: Erlbaum.

Calvo, R. A., & D'Mello, S. K. (2010). Affect detection: An interdisciplinary review of models, methods, and their applications. *IEEE Transactions on Affective Computing, 1,* 18-37.

Cohen, P. A., Kulik, J. A., & Kulik, C. C. (1982). Educational outcomes of tutoring: A meta-analysis of findings. *American Educational Research Journal, 19,* 237-248.

Csikszentmihalyi, M. (1990). *Flow: The psychology of optimal experience.* New York: Harper-Row.

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. (2012). AutoTutor and affective AutoTutor: Learning by talking with cognitively and emotionally intelligent computers that talk back. *ACM Transactions on Interactive Intelligent Systems, 2*(23), 1-38.

D'Mello, S., Lehman, S., Pekrun, R., & Graesser, A. (2014). Confusion can be beneficial for learning. *Learning and Instruction, 29,* 153-170.

Dodds, P. V. W., & Fletcher, J. D. (2004). Opportunities for new "smart" learning environments enabled by next generation web capabilities. *Journal of Education Multimedia and Hypermedia, 13,* 391-404.

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

Dynarsky, M., Agodina, R., Heaviside, S., Novak, T., Carey, N., Campuzano, L., ... Sussex, W. (2007). *Effectiveness of reading and mathematics software products: Findings from the first student cohort.* Washington, DC: U.S. Department of Education, Institute of Education Sciences.

Engineering and Computing Simulations. (2012). *vMedic*. Retrieved October 2, 2013 from www.ecsorl.com/products/vmedic

Fletcher, J. D. (2003). Evidence for learning from technology-assisted instruction. In H. F. O'Neil & R. S. Perez (Eds.), *Technology applications in education: A learning view* (pp. 79–99). Mahwah, NJ: Erlbaum.

Fletcher, J. D. (2009). Education and training technology in the military. *Science, 323,* 72-75.

Fletcher, J. D. (2014). *Digital tutoring in information systems technology for veterans: Data report* (Document D-5336). Alexandria, VA: Institute for Defense Analyses.

Fletcher, J. D., & Morrison, J. E. (2012). *DARPA Digital Tutor: Assessment data* (IDA Document D-4686). Alexandria, VA: Institute for Defense Analyses.

Forsyth, C. M., Pavlik, P., Graesser, A. C., Cai, Z., Germany, M., Millis, K., … & Dolan, R. (2012). Learning gains for core concepts in a serious game on scientific reasoning. In K. Yacef, O. Zaïane, H. Hershkovitz, M. Yudelson, & J. Stamper (Eds.), *Proceedings of the 5th International Conference on Educational Data Mining* (pp. 172-175). Chania, Greece: International Educational Data Mining Society.

Graesser, A. C., Chipman, P., Leeming, F., & Biedenbach, S. (2009). Deep learning and emotion in serious games. In U. Ritterfeld, M. Cody, & P. Vorderer (Eds.), *Serious games: Mechanisms and effects* (pp. 81-100). New York and London: Routledge, Taylor & Francis.

Graesser, A. C., Chipman, P., Haynes, B., & Olney, A. (2005). AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education, 48,* 612-618.

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., & Cade, W. (2011). Instruction based on tutoring. In R. E. Mayer & P. A. Alexander (Eds.), *Handbook of Research on Learning and Instruction* (pp. 408-426). New York: Routledge Press.

Graesser, A. C., & D'Mello, S. (2012). Emotions during the learning of difficult material. In. B. Ross (Ed.), *The Psychology of Learning and Motivation* (Vol. 57, pp. 183-225). Amsterdam, Netherlands: Elsevier.

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., & King, B. (2008). Technology-based training. In J. J. Blascovich & C. H. Hartel (Eds.), *Human behavior in military contexts* (pp. 127-149). Washington, DC: National Academy of Sciences.

Graesser, A. C., Lu, S., Jackson, G. T., Mitchell, H., Ventura, M., Olney, A., & Louwerse, M. M. (2004). AutoTutor: A tutor with dialogue in natural language. *Behavioral Research Methods, Instruments, and Computers, 36,* 180-193.

Halpern, D. F., Millis, K., Graesser, A. C., Butler, H., Forsyth, C., & Cai, Z. (2012). Operation ARA: A computerized learning game that teaches critical thinking and scientific reasoning. *Thinking Skills and Creativity, 7,* 93-100.

Jackson, G. T., Dempsey, K. B., & McNamara, D.S. (in press). Game-based practice in reading strategy tutoring system: Showdown in iSTART-ME. In H. Reinders (Ed.), *Computer games.* Bristol, UK: Multilingual Matters.

Jackson, G. T., & Graesser, A. C. (2007). Content matters: An investigation of feedback categories within an ITS. In R. Luckin, K. Koedinger, & J. Greer (Eds.), *Artificial intelligence in education: Building technology rich learning contexts that work* (pp. 127–134). Amsterdam: IOS Press.

Jackson, G. T., & McNamara, D. S. (2013). Motivation and performance in a game-based intelligent tutoring system. *Journal of Educational Psychology, 105,* 1036-1049.

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

Jurafsky, D., & Martin, J. H. (2008). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.* Upper Saddle River, NJ: Prentice-Hall.

Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education, 8,* 30-43.

Koedinger, K. R., Corbett, A. C., & Perfetti, C. (2012). The Knowledge-Learning-Instruction (KLI) framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science, 36*(5), 757-798.

Landauer, T. K., Laham, D., & Foltz, P. W. (2003). Automatic essay assessment. *Assessment in Education: Principles, Policy & Practice, 10,* 295-308.

Landauer, T. K., McNamara, D. S., Dennis, S., & Kintsch, W. (Eds.). (2007). *Handbook of latent semantic analysis.* Mahwah, NJ: Erlbaum.

Lepper, M. R., & Henderlong, J. (2000). Turning "play" into "work" and "work" into "play": 25 years of research on intrinsic versus extrinsic motivation. In C. Sansone & J. M. Harackiewicz (Eds.), *Intrinsic and extrinsic motivation: The search for optimal motivation and performance* (pp. 257-307). San Diego, CA: Academic Press.

Lesgold, A., Lajoie, S. P., Bunzo, M., & Eggan, G. (1992). SHERLOCK: A coached practice environment for an electronics trouble-shooting job. In J. H. Larkin & R. W. Chabay (Eds.), *Computer assisted instruction and intelligent tutoring systems: Shared goals and complementary approaches* (pp. 201–238). Hillsdale, NJ: Erlbaum.

Ma, W., Adesope, O. O., & Nesbit, J. C. (in press). Intelligent tutoring systems and learning outcomes: A meta-analytic survey. *Journal of Educational Psychology.*

Mayer, R. E. (2011). Multimedia learning and games. In S. Tobias & J. D. Fletcher (Eds.), *Computer games and instruction* (pp. 281–305). Charlotte, NC: Information Age.

McCarthy, P., & Boonthum-Denecke, C. (Eds.). (2012). *Applied natural language processing: Identification, investigation, and resolution.* Hershey, PA: IGI Global.

McNamara, D. S., Jackson, G. T., & Graesser, A. C. (2010). Intelligent tutoring and games (ITaG). In Y. K. Baek (Ed.), *Gaming for classroom-based learning: Digital role-playing as a motivator of study* (pp. 44-65). Hershey, PA: IGI Global.

McQuiggan, S. W., Robison, J. L., & Lester, J. C. (2010). Affective transitions in narrative-centered learning environments. *Educational Technology & Society, 13,* 40–53.

Millis, K., Forsyth, C., Butler, H., Wallace, P., Graesser, A., & Halpern, D. (2011). Operation ARIES! A serious game for teaching scientific inquiry. In M. Ma, A. Oikonomou, & J. Lakhmi (Eds.), *Serious games and edutainment applications* (pp.169-196). London, UK: Springer-Verlag.

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

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

Nye, B. D., Graesser, A. C., & Hu, X. (2014). AutoTutor and family: A review of 17 years of natural language tutoring. *International Journal of Artificial Intelligence in Education, 24.*427-469.

Nye, B. D., Hu, X., Graesser, A. C., & Cai, Z. (in press). AutoTutor in the Cloud: A service-oriented paradigm for an interoperable natural language ITS. *Journal of Advanced Distributed Learning Technology.*

Nye, B. D., Sottilare, R. A., Ragusa, C., & Hoffman, M. (2014). Defining instructional challenges, strategies, and tactics for adaptive intelligent tutoring systems. In R. A. Sottilare, A. C. Graesser, X. Hu, & B. Goldberg (Eds.), *Design Rrecommendations for Iintelligent Ttutoring Ssystems: Instructional Management* (Vol. 2, pp. xv-xxvi). Orlando, FL: Army Research Laboratory.

O'Neil, H. F., & Perez, R. S. (Eds.). (2008). *Computer games and team and individual learning.* Amsterdam, Netherlands: Elsevier.

Olney, A., D'Mello, S. K., Person, N., Cade, W., Hays, P., Williams, C., … & Graesser, A. C. (2012). Guru: A computer tutor that models expert human tutors. In S. Cerri, W. Clancey,

G. Papadourakis, & K. Panourgia (Eds.), *Proceedings of Intelligent Tutoring Systems (ITS) 2012* (pp. 256-261). Berlin, Germany: Springer.

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

Ritterfeld, U., Cody, M., & Vorderer, P. (Eds.). (2009). *Serious games: Mechanisms and effects*. New York and London: Routledge, Taylor & Francis.

Rowe, J. P., Shores, L. R., Mott, B. W., & Lester, J. C. (2011). Integrating learning, problem solving, and engagement in narrative-centered learning environments. *International Journal of Artificial Intelligence in Education*, *21,* 115–133.

Rus, V., D'Mello, S., Hu, X., & Graesser, A. C. (2013). Recent advances in intelligent systems with conversational dialogue. *AI Magazine, 34*, 42-54.

Sabourin, J. L., Rowe, J. P., Mott, B. W., & Lester, J. C. (2013). Considering alternate futures to classify off-task behavior as emotion self-regulation: A supervised learning approach. *Journal of Educational Data Mining, 5*, 9-38.

Shute, V. J., & Ventura, M. (2013). *Measuring and supporting learning in games: Stealth assessment.* Cambridge, MA: MIT Press.

Sottilare, R. A., Goldberg, B. S., Brawner, K. W., & Holden, H. K. (2012). A modular framework to support the authoring and assessment of adaptive computer-based tutoring systems (CBTS). In *Proceedings of the Interservice/Industry Training, Simulation & Education Conference (I/ITSEC) 2012* (12017). Arlington, VA: National Training Systems Association.

Sottilare, R., Graesser, A., Hu, X., & Goldberg, B. (Eds.). (2014). *Design Recommendations for Intelligent Tutoring Systems: Instructional Management* (Vol. 2). Orlando, FL: Army Research Laboratory.

Sottilare, R., Graesser, A., Hu, X., & Holden, H. (Eds.). (2013). *Design Recommendations for Intelligent Tutoring Systems: Learner Modeling* (Vol. 1). Orlando, FL: Army Research Laboratory.

Steenbergen-Hu, S., & Cooper, H. (2013). A meta-analysis of the effectiveness of intelligent tutoring systems on K-12 students' mathematical learning. *Journal of Educational Psychology, 105*, 971-987.

Steenbergen-Hu, S., & Cooper, H. (2014). A meta-analysis of the effectiveness of intelligent tutoring systems on college students' academic learning. *Journal of Educational Psychology, 106*, 331-347.

Tobias, S., & Fletcher, J. D. (2011). *Computer games and instruction.* Charlotte, NC: Information Age.

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

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

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., Jordan, P., Rosé, C. P., Bhembe, D., Böttner, M., Gaydos, A., … Srivastava, R. (2002). The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In

S. A. Cerri, G. Gouarderes, & F. Paraguacu (Eds.), *Intelligent Tutoring Systems: 6th International Conference* (pp. 158-167). Berlin: Springer.

Ward, W., Cole, R., Bolaños, D., Buchenroth-Martin, C., Svirsky, E., & Weston, T. (2013). My science tutor: A conversational multimedia virtual tutor. *Journal of Educational Psychology, 105*, 1115-1125.

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

Wouters, P., van Nimwegen, C., van Oostendorp, H., & van der Spek, E. D. (2013). A meta-analysis of the cognitive and motivational effects of serious games. *Journal of Educational Psychology, 105*, 249-265.

Author Notes

The research was supported by the National Science Foundation (SBR 9720314, REC 0106965, REC 0126265, ITR 0325428, REESE 0633918, ALT-0834847, DRK-12-0918409, 1108845), the Institute of Education Sciences (R305H050169, R305B070349, R305A080589, R305A080594, R305G020018, R305C120001), Army Research Lab (W911INF-12-2-0030), and the Office of Naval Research (N00014-00-1-0600, N00014-12-C-0643). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF, IES, or DoD. The Tutoring Research Group (TRG) is an interdisciplinary research team comprised of researchers from psychology, computer science, physics, and education at University of Memphis (visit http://www.autotutor.org, http://emotion.autotutor.org, http://fedex.memphis.edu/iis/).

Table 4.1

High-Level Summary of GIFT Strategy Evaluation

Step	Module	Description	Functional Expression
1.A	Communication	The learner interacts with user interface and game environment, which sends events to the domain module	$\mathbf{E_t}$ – Events from user behavior at time **t**
1.B	Sensor	Sensor data states (e.g., emotion classifications) are sent directly to Learner Module	$\mathbf{D_t}$ – Data from sensors at time **t**
1.C	Learner	Persistent learner model data, such as the contents of a learning management system	$\mathbf{D_L}$ – Persistent learner model data
2	Domain	Performance assessment rules estimate discrete performance (e.g., below, at, or above expectation) on low-level domain concepts ($\mathbf{C_L}$)	$\mathbf{P_{c,t}}$ – Performance on concept **c** at time **t** $\mathbf{R_c}$ = Rules to assess **c** $\mathbf{P_{c,t}} = \mathbf{R_c}(\mathbf{E_t}) \quad \forall\ \mathbf{c} \in \mathbf{C_L}$
3	Domain	Performance for higher level concepts ($\mathbf{C_H}$) "rolled up" (aggregated) from lower levels	$\mathbf{F_c}$ – Roll-up function for **c** $\mathbf{P_{c,t}} = \mathbf{F_c}(\mathbf{C_L}) \quad \forall\ \mathbf{c} \in \mathbf{C_H}$
4	Domain	Performance assessment states sent to pedagogical module and to learner module	$\mathbf{P_t}$ – Performance states for all concepts at **t** $\mathbf{P_t} = [\mathbf{P_{c1,t}}, \mathbf{P_{c2,t}}, \ldots]$

5	Learner	Learner state changes on domain performance (Step 5) or sensors (Step 1.B) trigger strategy evaluation	
6	Pedagogy	Instructional strategy selected based on the transition from the prior to current learner state	$\mathbf{S_t}$ – Strategy for time **t** $\mathbf{F_S}$ – Strategy selection code for pedagogy module $\mathbf{S_t = F_S(P_{t-1}, D_{t-1}, P_t, D_t, D_L)}$
7	Pedagogy	Strategy selection is sent to the domain module	
8	Domain (Tactics)	A strategy selection is mapped to a domain-specific tactic (**T**)	$\mathbf{T_t}$ – Tactic selected $\mathbf{T_t = F_T(S_t)}$
9	Communication	A tactic causes one or more actions (**A**) to occur to the game environment (e.g., hints, changes in difficulty, etc.)	$\mathbf{A_t}$ – Actions for time **t** $\mathbf{F_A}$ – Map of tactics (**T**) to environment actions (**A**) $\mathbf{A_t = F_A(T)}$