AutoTutor

Arthur Graesser, Sidney D'Mello, Xiangen Hu, Zhiqiang Cai, Andrew Olney, and Brent Morgan
University of Memphis, Memphis, Tennessee USA

ABSTRACT

AutoTutor is an intelligent tutoring system that helps students learn science, technology, and other technical subject matters by holding conversations with the student in natural language. AutoTutor's dialogues are organized around difficult questions and problems that require reasoning and explanations in the answers. The major components of AutoTutor include an animated conversational agent, dialogue management, speech act classification, a curriculum script, semantic evaluation of student contributions, and electronic documents (e.g., textbook and glossary). This chapter describes the computational components of AutoTutor, the similarity of these components to human tutors, and some challenges in handling smooth dialogue. We describe some ways that AutoTutor has been evaluated with respect to learning gains, conversation quality, and learner impressions. AutoTutor is sufficiently modular that the content and dialogue mechanisms can be modified with authoring tools. AutoTutor has spawned a number of other agent-based learning environments, such as AutoTutor-lite, Guru, and Operation Aries!.

INTRODUCTION AND BACKGROUND

Intelligent Tutoring Systems (ITS) are computerized learning environments that incorporate computational models in the cognitive sciences, learning sciences, artificial intelligence, computational linguistics, and other fields that develop intelligent systems (Sleeman & Brown, 1982; Woolf, 2009). In a process called student modelling, the ITS tracks the psychological states of learners, such as subject matter knowledge, cognitive skills, strategies, motivation, and emotions. An ITS adaptively responds with activities that are sensitive to these psychological states, the history of the student-tutor interaction, and the instructional agenda. An ITS is very different from more rigid, insensitive, and inflexible learning environments such as reading a book or listening to a lecture.

ITS environments were originally developed for mathematically well-formed subject matters. Impressive systems have been developed and tested for algebra, geometry, and programming languages (the *Cognitive Tutors*: Anderson et al., 1995; Koedinger et al., 1997; Ritter et al., 2007, *ALEKS*: Doignon & Falmagne, 1999), for physics (*Andes, Atlas*: VanLehn et al., 2002), for electronics (*SHERLOCK*: Lesgold, Lajoie, Bunzo, & Eggan, 1992), and for information technology (*KERMIT*: Mitrovic, Martin, & Suraweera, 2007). More recently the ITS enterprise has evolved to handle conversational interaction in natural language on verbal topics that require conceptual reasoning. This chapter focuses on *AutoTutor* (Graesser, Lu et al., 2004), but other systems have been developed with similar goals: *ITSPOKE* (Litman et al., 2006), *Spoken Conversational Computer* (Pon-Barry, Clark, Schultz, Bratt, Peters, & Haley, 2005), *Tactical Language and Culture Training System* (Johnson & Valente, 2008), *Why-Atlas* (VanLehn et al., 2007), and iSTART (McNamara, Levinstein, & Boonthum, 2004). These systems automatically analyze language and discourse by incorporating recent advances in computational linguistics (Jurafsky & Martin, 2008) and statistical representations of world knowledge (Landauer, McNamara, Dennis, & Kintsch, 2007).

Most ITS fit within VanLehn's (2006) analyses of the outer loop and the inner loop when characterizing the scaffolding of solutions to problems, answers to questions, or completion of complex tasks. The outer loop involves the selection of topics and problems to cover, assessments of the student's topic knowledge and general cognitive abilities, and global aspects of the tutorial interaction. The inner loop consists of covering individual steps within a problem at a micro-level. Adaptivity and intelligence are necessary at both the outer loop and the inner loop in a bona fide ITS.

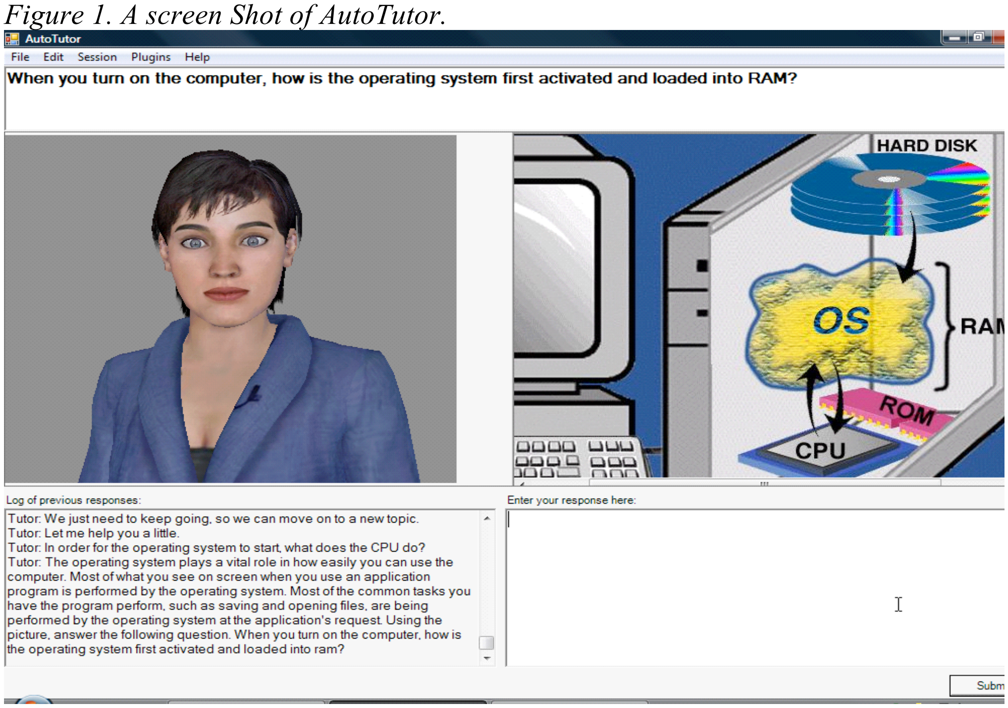
This chapter describes the computational components of AutoTutor and some of the challenges faced when simulating smooth and pedagogically effective dialogue. AutoTutor's architecture incorporates dialogue mechanisms of human tutors in addition to ideal tutoring strategies. We describe evaluations of AutoTutor with respect to learning gains, conversation quality, and learner impressions. The modular architecture of AutoTutor allows developers to develop new content and dialogue strategies with authoring tools. We end the chapter by identifying some of AutoTutor's progeny that also have conversational agents, such as AutoTutor-lite, Guru, and Operation Aries!.

AUTOTUTOR MECHANISMS

AutoTutor simulates a tutor by holding a conversation in natural language (Graesser, Chipman, Haynes, & Olney, 2005; Graesser, Jeon, & Dufty, 2008; Graesser, Graesser, Lu et al., 2004; Graesser, Person, & Harter, 2001). Students type in their contributions through a keyboard in most applications. However, we have developed a version that handles spoken input from the student through the Dragon Naturally Speaking ™ (version 6) speech recognition system (D'Mello, King, Chipman, & Graesser, in press). AutoTutor communicates through an animated conversational agent with speech, facial expressions, and some rudimentary gestures.

Figure 1 shows a screen shot of AutoTutor on the topic of computer literacy. Most versions of AutoTutor have the three major areas shown in Figure 1. Area 1 (top of screen) is the main question (or problem) that stays on the computer screen throughout the conversation that collaboratively constructs an answer to the question. Area 2 (left middle) is the animated conversational agent that speaks the content of AutoTutor's turns. Area 3 (right middle) is either blank or has auxiliary diagrams on the subject matter. When the students type in their contributions, there is an area at the bottom that displays what the student types in. In versions with speech recognition, there are two buttons on the keyboard that the learner presses to start speaking and stop speaking. The interface can also include a dialogue area that presents the history of the turn-by-turn tutorial dialogue; students can scroll back as far as they want in this dialogue history.

Figure 1. A screen Shot of AutoTutor.



The outer loop of AutoTutor consists of a serious of didactic lessons and challenging problems or questions (such as *why, how, what-if*). The example main question in Figure 1 is "When you turn on the computer, how is the operating system first activated and loaded into RAM?" The order of lessons, problems, and questions can be dynamically selected based on the profile of student abilities, but the order is fixed in most versions of AutoTutor we have developed. The interactive dialogue occurs during the problems/questions but not the didactic delivery of information (e.g., reading text, viewing a diagram). The answer to a question (or solution to a problem) requires several sentences of information in an ideal answer. AutoTutor assists the learner in constructing their answer after the student enters their initial response. The inner loop of AutoTutor consists of this collaborative interaction while answering a question (or solving a problem). It is this inner loop that is the distinctive hallmark of AutoTutor. The tutor draws out more of the student's knowledge (through hints and prompts), helps fill in missing information, repairs misconceptions, and answers student questions. The inner loop dialogue between AutoTutor and the student takes approximately 100 dialogue turns to answer a single challenging question, approximately the length of a conversation with a human tutor.

There are augmented versions of AutoTutor with special features that are designed to enhance the learning experience. Several versions of AutoTutor have been developed since 1997, when the system was created. One version of AutoTutor, called *AutoTutor-3D,* guides learners on using interactive simulations of physics microworlds (Graesser, Chipman et al., 2005; Jackson et al., 2006). For each of the physics problems, there is an interactive simulation world with people, objects, and the spatial setting associated with the problem. The student manipulates parameters of the situation (e.g., mass of objects, speed of objects, distance between objects), asks the system to simulate what will happen, and then gets prompted by the tutor to describe what the student sees. AutoTutor manages the dialogue with hints and suggestions that scaffold the learning process with dialogue. Another version is sensitive to the student's emotions, which are tracked automatically by detecting particular dialogue patterns, facial expressions, and body motions (D'Mello & Graesser, in press; chapter XX in this volume). Yet another version adds game features with feedback and points that are contingent on the student's performance (Jackson & Graesser, 2007). However, it is the conversational interaction of AutoTutor's inner loop that is the signature contribution of AutoTutor.

What does AutoTutor do?

Perhaps the best way to convey what AutoTutor does in the inner loop is through an example conversational dialogue. The excerpts below were extracted from an actual conversation with a student on the topic of conceptual physics. The conversation begins with a main question selected by AutoTutor.

AutoTutor: *When a car without headrests on the seats is struck from behind, the passengers often suffer neck injuries? Explain why.* (MAIN QUESTION)

Student: *I don't know.* (This expression by the student is classified as a Metacognitive speech act because it conveys what the student knows about his/her own knowledge.)

AutoTutor: *Could you at least try to give me an answer? That way, we'll have something as a starting place to work from.* (PUMP. This dialogue move encourages the student to say more, but does not direct the student to express particular information.)

As the dialogue continues, AutoTutor detects that an important expected piece of information is missing (i.e., the impact of the collision on the neck).

AutoTutor: *Let's try this together.* (DISCOURSE MARKER. This is a discourse marker that introduces or connects dialogue moves within a turn.)

When the collision provides a large force to accelerate the head, what could happen to the neck? (HINT. This dialogue move leads the student to articulate a particular sentence, clause, or proposition.)

At one point in the dialogue, the student asks a question, so AutoTutor needs to provide an answer. AutoTutor can answer some classes of questions (such as definitional questions), but not all questions, so AutoTutor needs to productively manage the dialogue surrounding student question asking.

Student: *What is a force?*
AutoTutor: *I'll take a crack at answering your question.* (DISCOURSE MARKER)
Force is any influence that tends to accelerate an object; a push or pull; measured in newtons. Force is a vector quantity. (ANSWER. AutoTutor answers the student's direct question. Answers are either accessed from electronic glossaries or a paragraph is dynamically selected from a textbook.)
Was that what you were looking for? (QUESTION SOLICITATION. AutoTutor allows the student the opportunity to rephrase his question.)

As the dialogue continues AutoTutor believes it has detected a misconception.

AutoTutor: *See if you know the answer to this.* (DISCOURSE MARKER. AutoTutor prepares the student for a direct question.)
When the head and body are moving at the same acceleration, they are moving at the same ________? (PROMPT. This dialogue move encourages the student to fill in a particular word or phrase of information, in this case the word "rate." Prompts are frequently needed for students who have low verbal fluency in order to elicit a particular noun, verb, or adjective that is missing in an answer.)
Student: *distance*
AutoTutor: *Moving at the same rate.* (PROMPT COMPLETION. AutoTutor splices in the correct information if the student provides an incorrect answer to its prompt.)

At this point, the dialogue continues until all of the relevant expected answers are covered in the conversation. The conversation then ends with a summary answer by the tutor.

The example illustrates the mixed-initiative dialogue of AutoTutor. AutoTutor attempts to interpret or otherwise handle any question, assertion, comment, or extraneous speech acts that the student expresses. Each turn of AutoTutor requires the generation of one or more dialogue moves that adaptively respond to what the student expresses, that advance the conversation in a constructive manner, that covers good answer information, or that corrects misconceptions. The tutor's dialogue moves within a turn are connected by discourse markers, as illustrated in the example. Some dialogue moves are responsive to the student's preceding turn, such as short feedback (positive, neutral, versus negative), answers to student questions, and corrections of student misconceptions. Other dialogue moves push the dialogue forward in an attempt to cover the expected answers to the main question.

Dialogue Structure

The dialogue structure of AutoTutor is similar to the dialogue patterns of human tutors. Graesser and Person analyzed dialogue patterns of typical human tutors in middle school and in college (Graesser & Person, 1994; Graesser, Person, & Magliano, 1995). Similar analyses have been conducted by other researchers on naturalistic tutoring corpora (Chi et al., 2004; Evens & Michael, 2006; Litman et al., 2006). The following dialogue structures are implemented in AutoTutor and are prominent in human tutors: (a) a curriculum script with didactic content and problems (i.e., difficult tasks or questions), (b) a

5-step Tutoring Frame, (c) Expectation and Misconception Tailored (EMT) dialogue, and (d) Conversational Turn Management.

(a) **Curriculum script**. The tutor covers a curriculum with didactic content and a set of questions or problems that address the content. Didactic content can be presented in a mini-lecture, hopefully at the appropriate time for each individual learner. The questions/problems require the student to actively apply their knowledge. The curriculum script includes expected answers, misconceptions, hints, prompt questions, and other inner loop information.

(b) **5-Step Tutoring Frame.** When a challenging main question (or problem) is selected to work on, the question is answered through an interaction that is structured by a 5-Step Tutoring Frame. The 5 steps are: (1) Tutor presents a main question, (2) Student gives an initial answer, (3) Tutor gives short feedback on the quality of the Student's initial answer, (4) the Tutor and Student collaboratively improve on the answer in a turn-by-turn dialogue that may be lengthy, and (5) the Tutor evaluates whether the Student understands (e.g., asking "Do you understand?" or testing with a follow-up task). This 5-step tutoring frame involves collaborative discussion, joint action, and encouragement for the student to construct knowledge rather than merely receiving knowledge.

(c) **Expectation and Misconception Tailored (EMT) Dialogue**. Human tutors typically have a list of *expectations* (i.e. anticipated good answers or steps in a procedure) and a list of anticipated *misconceptions* (incorrect information) associated with each main question. They want the expectation content covered in order to handle the main question that is selected. The tutor guides the student in articulating the expectations through a number of dialogue moves, namely *pumps* ("What else?"), *hints*, *prompt questions* to extract specific information from students, *assertions* that capture particular expectations, and *answers* to students' questions. As the dialogue progresses, tutors tend to lead more while trying to get the student to articulate an expectation. They start with a pump and then move to a hint if the pump fails, followed by a prompt question and an assertion if students fail to articulate the expectation. The pump → hint → prompt → assertion cycle is implemented by AutoTutor to encourage the student to articulate the answer and cover expectations. The correct answers are eventually covered and the misconceptions are hopefully corrected.

(d) **Conversational Turn Management.** Human tutors structure their conversational turns systematically. Nearly every turn of the tutor has three information slots. The first slot of most turns is feedback on the quality of the learner's last turn. This feedback is either positive (*very good, yeah*), neutral (*uh huh, I see*), or negative (*not quite, not really*). The second slot advances the interaction with a prompt for specific information, a hint, an assertion with correct information, a correction of misconceptions, or an answer to the student's question. The third slot is a cue for the floor to shift from the tutor as the speaker to the learner. For example, the human ends each turn with a question or a gesture to cue the learner to do the talking. Otherwise the student and AutoTutor are at a standstill waiting for the other to take the next turn.

Student Modeling

One of the central questions is how well the tutor can track the psychological states of the student as the tutor implements tutoring strategies. Available evidence suggests that human tutors are not able to conduct student modeling at a fine-grained level (Chi, Siler, & Jeong, 2004; Graesser, D'Mello, & Person 2009). They are limited to performing approximate assessments rather than fine-grain assessments. Computers can potentially show advantages over humans to the extent that artificial intelligence can accurately conduct student modeling and generate intelligent responses.

Student modeling in the inner loop consists of comparing what the student express in language with the list of expectations and misconceptions associated with a main question. For example, supposed that expectations E1 and E2 and misconceptions M1 and M2 are relevant to a particular physics question that involves a head-on collision between a large and small vehicle.

E1. The magnitudes of the forces exerted by A and B on each other are equal.
E2. If A exerts a force on B, then B exerts a force on A in the opposite direction.

M1: A lighter/smaller object exerts no force on a heavier/larger object.

M2: Heavier objects accelerate faster for the same force than lighter objects

AutoTutor guides the student in articulating the expectations through pumps, hints, and prompts. Hints and prompts are carefully selected by AutoTutor to produce content in the answers that fills in missing content words, phrases, and propositions. For example, a hint to get the student to articulate expectation E1 might be “What about the forces exerted by the vehicles on each other?”; this hint would ideally elicit the answer “The magnitudes of the forces are equal.” A prompt to get the student to say “equal” would be “What are the magnitudes of the forces of the two vehicles on each other?” If the student fails to articulate E1 after many attempts, then AutoTutor asserts the expectation at the end of the pump → hint → prompt → assertion cycle. However, there is an early exit from the cycle when the student articulates the information in E1. As the learner expresses information over many turns, the list of expectations is eventually covered and the main question is scored as answered.

Complete coverage of the answer requires AutoTutor to have a pool of hints and prompts in the curriculum script that are available to extract all of the content words, phrases, and propositions in each expectation. AutoTutor adaptively selects those hints and prompts that fill missing constituents and thereby achieves *pattern completion*. For example, the following family of candidate prompts is available for selection by AutoTutor to encourage the student to articulate words in expectation E1.

(a) The magnitudes of the forces exerted by two objects on each other are ____.
(b) The magnitudes of forces are equal for the two ______.
(c) The two vehicles exert on each other an equal magnitude of _____.
(d) The force of the two vehicles on each other are equal in _____.

If the student has failed to articulate one of the four content words (*equal, objects, force, magnitude*), then AutoTutor selects the corresponding prompt (a, b, c, and d, respectively).

Student modelling is executed after every student turn by comparing the verbal contributions of the student with the list of expectations and misconceptions. This requires *semantic matching* algorithms that compare the student input with AutoTutor’s Es and Ms. However, it is widely acknowledged that natural language is imprecise, fragmentary, vague, ungrammatical, and elliptical, so it would not be prudent to rely entirely on semantically well-formed semantic matches. AutoTutor therefore incorporates several semantic evaluation algorithms when performing these matches, but notably Latent Semantic Analysis (Landauer et al., 2007), regular expressions (Jurafsky & Martin, 2008), content word overlap metrics (that have higher weight for low frequency words than high frequency words), and occasionally logical entailment (Rus & Graesser, 2006).

As an example, early versions of AutoTutor relied exclusively on LSA in its semantic evaluation of student input. The LSA algorithm computed the extent to which the information within the student turns (i.e., an individual turn, a combination of turns, or collective sequence of turns) matches each expectation in the ideal answer. Expectation E_i is considered covered if the content of the learner’s cumulative set of turns meets or exceeds a threshold T in its LSA cosine value (which varies from near 0 to 1). That is, E_i is covered if the cosine match between E_i and the student input I (including turns 1 though N) is high enough: cosine $(E_i, I) \geq T$. The threshold has varied between .40 and .75 in previous instantiations of AutoTutor. Each expectation E_i has an associated family of prompts and hints to get the student to fill in most or all of the content words and propositions in E_i. Prompts and hints are selected to maximize an increase in the LSA cosine match score (hereafter called the *match score*) when answered successfully. Stated differently, hints and prompts are selected to maximize pattern completion. Sometimes the student expresses misconceptions during the dialogue. This happens when the student input I matches a misconception M with a sufficiently high match score. At that point AutoTutor corrects the misconception and goes on.

During the course of the dialogue and student modeling, the system periodically identifies a missing expectation and posts the goal of covering the expectation. When expectation E_i is missed (and therefore posted), AutoTutor attempts to get the student to articulate it by generating hints and prompts affiliated with E_i to encourage the student to fill in missing words and propositions. The selection of the next E_i to cover follows the principle of the *zone of proximal development* or what some call frontier learning:

AutoTutor builds on what the student has managed to articulate. More formally, AutoTutor selects the next E_i from the set of expectations that (a) has the highest match score and (b) has a subthreshold match score. This *subthreshold expectation selection* algorithm assumes that the expectations should not be covered in a prescribed sequential order. However, ordering constraints may also be considered in a *sequential expectation selection* algorithm. Some subject matters have ordering constraints but others do not.

Sometimes there are errors in AutoTutor's semantic matching. This can be disconcerting to the student when the students believe they have provided good, relevant contributions, yet it seems AutoTutor is not listening, or the short feedback is negative. The AutoTutor research teams have spent considerable efforts in improving the semantic match algorithms with techniques that go beyond LSA (see Graesser, Penumatsa, Ventura, Cai, & Hu, 2007; Rus & Graesser, 2006), but it is beyond the scope of this chapter to describe these improvements. AutoTutor's feedback and dialogue move generator also face limitations when the curriculum script does not have a full family of hints and prompts to cover all of the content words in each expectation E_i. A lazy lesson planner or knowledge engineer may cut corners and fail to specify prompts for important content words. When this happens, there is the risk of the threshold T never being met in the semantic match computations, which spawns two unfortunate consequences: AutoTutor generates irrelevant or redundant prompts and hints, or AutoTutor generates assertions that echo what the student has already expressed (which seems like AutoTutor is not listening). AutoTutor works quite well, however, when the curriculum script is adequately constructed and the semantic matching algorithms are on the mark.

AutoTutor computes a number of metrics of psychological characteristics during the course of student modeling. These metrics are collected from information in a log file that records a rich amount of information about the AutoTutor-student interaction after each conversational turn. This chapter concentrates on the cognitive metrics whereas the chapter by D'Mello and Graesser (Chapter xx) covers metrics of emotions and motivation. The cognitive metrics vary in grain size and apply to either the inner loop, the outer loop, or both.

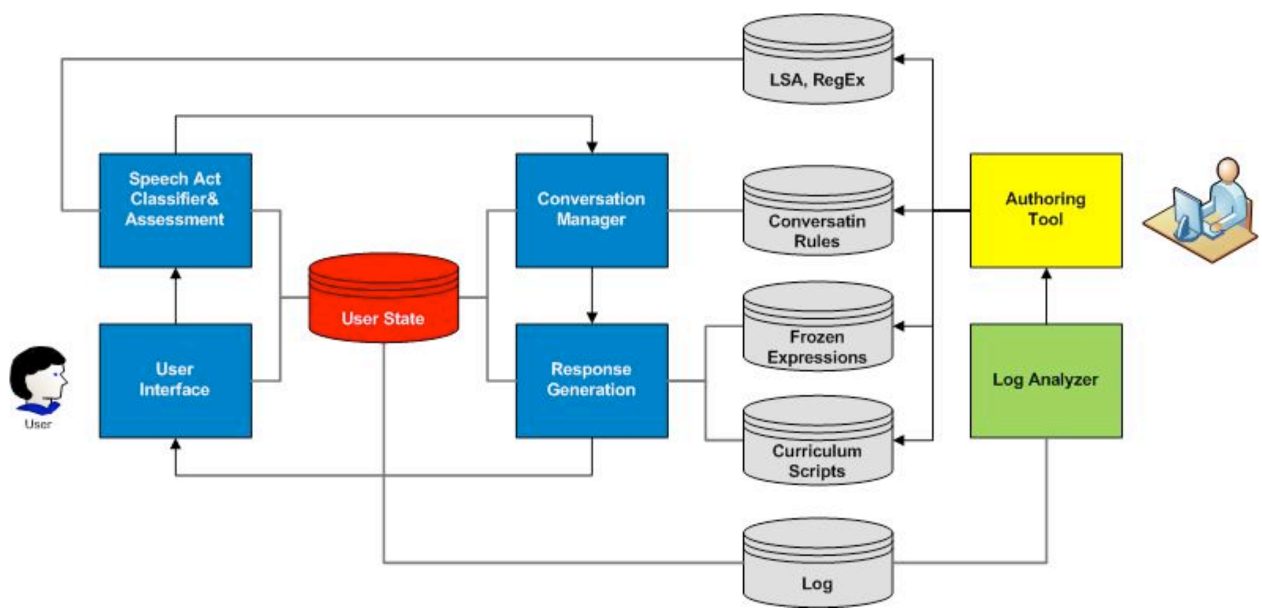
The quality of student contributions is computed at all levels of grain size: each turn, each expectation, each main question/problem, and the set of main questions in the curriculum script. The student modeling, therefore, is assessed from inner to outer loop, or local to global spans. The metric is similar at all levels. Specifically, the semantic match score computes what the student contributes compared with the expectations. Stated differently, does the student or AutoTutor have to articulate the content when answering the question? At the level of the turn, the match score is computed between the student's contribution and the expectation, with values varying from 0 to 1. A large number of scores are computed, including highest match score for all of the turns that address an expectation $E_{i,\ \text{and}}$ the mean match score over all the expectations for a main question. Global student knowledge for the subject matter is the mean match score when averaging over all of the previous questions/problems the student has worked on. Alternatively, relatively high match scores to misconceptions reflect low knowledge. Besides match scores, AutoTutor computes the volume of the student's contributions, which is called *verbosity*. A high-verbosity student expresses a large amount of information (measured in words or alphanumeric characters) compared with fellow students.

The accuracy of the student model algorithms have been evaluated over the years. In one analysis of conceptual physics, we collected pretest scores on a psychometrically validated test by Hestenes, Wells, and Swackhamer (1992), called the Force Concept Inventory. If AutoTutor is performing effective user modeling, then the dialogue moves selected by AutoTutor should be correlated with the students' prior knowledge of physics. Such predictions held up when we analyzed the dialogue moves of AutoTutor as a function of students of varying ability (Jackson & Graesser, 2006). For example, the short feedback that AutoTutor provides after the student's turns is either positive, neutral, or negative. The students' physics knowledge had a significant positive correlation with proportion of short feedbacks that were positive (r = .38) and a negative correlation with negative feedback (r = -.37). Another example applies to the corrections that AutoTutor made when identifying student errors and misconceptions. The correlation was negative (r = -.24), and marginally significant when compared with the corrections by AutoTutor. Yet another example considers the four dialogue move categories that attempt to cover the content of the expectations in the curriculum script: Pumps, hints, prompts, and assertions. The proportion of dialogue moves in these categories should be sensitive to the student's knowledge of physics. There is a continuum from the student-provided information to tutor-provided information as we move from pumps, to hints, to prompts, to assertions. The correlations with student knowledge reflected this continuum perfectly, with values of .49, .24, -.19, and -.40, respectively. Thus, for students with more knowledge of physics, all AutoTutor needs to do is primarily pump and hint, thereby encouraging or nudging the student to supply the answer to the question and articulate the expectations. For students with less knowledge of physics, AutoTutor needs to generate prompts for specific words or to assert the correct information, thereby extracting knowledge piecemeal or telling the student the correct information. These results support the claim that AutoTutor performs user modeling with some degree of accuracy and adaptively responds to the student's level of knowledge.

AutoTutor Architecture

Figure 2 presents the major components of AutoTutor's architecture. The bottom left depicts the student entering information via the user interface. The information in each student turn is segmented into speech acts, based on punctuation and (in some systems) a syntactic parser. Each speech act is assigned to one of approximately 20 speech act categories. These categories include assertions, 16 different categories of questions, short responses (yeah, right), meta-cognitive expressions (I don't understand, I see), and meta-communicative expressions (What did you say?). The accuracy of classifying the student speech acts into categories varies from .87-.96 (Olney et al., 2003), which is almost, but not quite, perfect. The dialogue coherence breaks down when some misclassification errors occur, which ends up confusing students. However, these problems are rare because the vast majority of student contributions are statement contributions or short responses, as opposed to questions. Students rarely take control in tutoring environments by asking questions, recommending problems to work on, or changing topics (Graesser, Person, & Magliano, 1995; Graesser, McNamara, & VanLehn, 2005). Instead, it is the tutor who controls the agenda.

Figure 2. Architecture of AutoTutor.



The speech acts expressed by the student on any given turn N constrain AutoTutor's conversation management of turn N+1. If the student asks a question, AutoTutor needs to answer it if it has an answer, or otherwise (a) generate dialogue moves to put the onus on the student to find an answer ("Good question. How would you answer it?") or (b) generate dialogue moves that evade getting an answer ("Good question, but I cannot answer it now. Let's move on."). If the student asks a metacognitive question ("I'm lost, I don't know"), which are normally frozen expressions, then AutoTutor acknowledges this and presents a hint to advance the dialogue in productive avenues. Student statement contributions are evaluated on quality, which drives the pump → hint → prompt → assertion cycles. The conversation dynamically flows from the student turns on the basis of the Conversation Manager module that is sensitive to the student's speech acts. The Conversation Manager module consists of a set of IF<state>THEN <action> production rules (Anderson & Gluck, 2001) or of a finite-state transition network (see Graesser, Person, & Harter, 2001). However, it is beyond the scope of this chapter to describe the computational mechanism of the Conversation Manager in more detail (see chapter xx by Olney & Graesser). The Conversation Manager subsequently passes information to the Response Generator, which is a sequence of dialogue moves and discourse markers. This content is expressed either in text or by an animated conversational agent that is displayed on the interface.

Animated conversational agents have become ubiquitous in recent advanced learning environments (Baylor & Kim, 2005; Gholson et al., 2009; McNamara, Levinstein, & Boonthum, 2004; Moreno & Mayer, 2007). The agents express themselves with speech, facial expression, gesture, posture, and other embodied actions. AutoTutor has used a wide array of agents that vary in quality of the speech and visual display. However, the learning gains of AutoTutor are not affected much by these agent characteristics (Graesser et al., 2003), whereas the impact of the dialogue content is extremely robust. Therefore, most of the AutoTutor team's efforts have concentrated on the content and conversation rather than the aesthetics of the talking head. We anticipate that the flashy dimensions of the animated agents will be developed by other research teams and eventually integrated with AutoTutor.

As depicted in Figure 2, AutoTutor has a repository of different static data structures that can be created and updated with authoring tools. First, all versions of AutoTutor represent world knowledge as *LSA* spaces (see Chapter xx by Kintsch & Kintsch), but some versions of AutoTutor or its progeny have incorporated other forms of world knowledge representation, such as textbooks, glossaries, and conceptual graph structures. Second, there are *Conversation Rules* that are represented as production rules, finite-state transition networks, or recursive augmented state transition networks (see chapter by Olney, Graesser & Person, chapter xx). Third, there are different categories of *Frozen Expressions* that have different discourse functions. For example, there are different ways for AutoTutor to express positive feedback (yes, yeah, good, great, fantastic, right on) and different ways that the student can express Metacommunicative speech acts (What did you say? Please repeat. I did not hear that.) Fourth, there is the *Curriculum Script*, as described earlier.

All of the information collected during the AutoTutor-student interaction is stored in the *Log* files. These files are fed into the *Log Analyzer* that can be inspected by the researcher and can inform the lesson planner or knowledge engineer who uses the *Authoring Tools*. These modules are, of course, standard for all learning management systems.

Evaluation of AutoTutor

AutoTutor has been evaluated on its psychological impact on the student. Perhaps the most important question is whether AutoTutor helps students learn. The learning gains of AutoTutor have been evaluated in over 20 experiments since its inception in 1997. Assessments of AutoTutor on learning gains have shown effect sizes of approximately 0.8 standard deviation units in the areas of computer literacy (Graesser et al., 2004) and Newtonian physics (VanLehn, Graesser et al., 2007). These effect sizes place AutoTutor somewhere between an untrained human tutor (Cohen, Kulik, & Kulik, 1982) and an intelligent tutoring system with ideal tutoring strategies (Corbett, 2001). AutoTutor improves learning between 0 and 2.1 sigma (a mean of 0.8), depending on the learning performance measure, the comparison condition, the subject matter, and the version of AutoTutor. Measures of learning have included: (1) multiple choice questions on shallow knowledge that tap definitions, facts and properties of concepts, (2) multiple choice questions on deep knowledge that taps causal reasoning, justifications of claims, and functional underpinnings of procedures, (3) essay quality when students attempt to answer challenging problems, (4) a cloze task that has students fill in missing words of texts that articulate explanatory reasoning on the subject matter, and (5) performance on problems that require problem-solving.

Assessments of learning gains obviously depend on the comparison conditions. The learning gains are approximately .8 for AutoTutor compared to a pretest and a condition of reading from a textbook on the same topics for an equivalent amount of time. The learning gains are approximately the same for AutoTutor and an expert human tutor who interacts with the student by computer-mediated communication (as opposed to face-to-face). The largest learning gains from AutoTutor have been on deep-reasoning measures rather than measures of shallow knowledge. AutoTutor is most effective when there is an intermediate gap between the student's prior knowledge and the ideal answers of AutoTutor; AutoTutor is not particularly effective in facilitating learning in students with high domain knowledge, nor when the material is too much over the student's head. It should be noted that the effectiveness of AutoTutor is less prominent in comparison conditions that attempt to control for the content that students are exposed to. The conversational AutoTutor has (a) a 0.22 sigma compared with reading textbook segments directly relevant to the AutoTutor's main questions/problems, (b) a 0.07 sigma compared with reading a script that succinctly answers the questions posed by AutoTutor, and (c) a 0.13 sigma compared with AutoTutor presenting speech acts in print instead of the talking head. The interactive AutoTutor-3D version has a .22 effect size over the normal conversational AutoTutor.

The conversations managed by AutoTutor are not perfect, but they are adequate for guiding students through the sessions with minimal difficulties. In fact, the dialogue is sufficiently tuned so that a bystander who observes tutorial dialogue in print cannot tell whether a particular turn was generated by AutoTutor or by an expert human tutor of computer literacy (Person & Graesser, 2002). A series of studies were conducted that randomly sampled AutoTutor's turns. Half of the turns were generated by AutoTutor and half were substituted by a human expert tutor on the basis of the dialogue history. Bystander participants were presented these tutoring moves in a written transcript and asked to decide whether each was generated by a computer or a human. The bystanders were unable correctly identify which moves were generated by the human tutor versus AutoTutor. Thus, AutoTutor successfully passed the *bystander Turing test* for individual tutoring turns. However, a bystander can eventually tell whether a sequence of turns was part of a dialogue with AutoTutor versus a human tutor. In conclusion, AutoTutor is close enough to human tutorial dialogue to keep the conversation going and also to promote learning.

Student ratings of AutoTutor have also been collected in order to get their impressions of the tutoring environment. The ratings lean toward the positive side, but there have been no systematic comparisons with human tutors or alternative learning environments. We have compared different versions of AutoTutor, but a provocative finding has made us somewhat skeptical about relying on ratings of student impressions. Specifically, Jackson and Graesser (2007) documented that there was a negative relationship between deep learning and enjoyment: students least preferred those versions from which they learned most. Students' metacognition of learning is limited (Graesser, D'Mello, & Person, 2009), so it is perhaps not surprising that their ratings of liking were not positively correlated with learning. Deep learning is challenging and sometimes painful, which may clash with an enjoyable experience for some groups of students.

These assessments point to the successes of AutoTutor, but it is important also to acknowledge some of its limitations. One limitation is that the conversational dialogue may have minimal incremental gains on learning when the exchange is time-consuming and the knowledge covered is shallow rather than deep. The conversational interaction is tedious for some students and even irritating far a small percentage. A second limitation is that students lose patience with AutoTutor when the conversation breaks down. As mentioned throughout this chapter, such breakdowns occur when the student modeling is imperfect, the curriculum script is incomplete, student speech acts are misclassified, and AutoTutor is viewed as being unresponsive to what the student is saying. A third limitation is that AutoTutor can correctly answer only a modest proportion of student questions (Graesser, McNamara, & VanLehn, 2005) so students eventually stop asking them. This puts a damper on self-regulated learning and also mixed-initiative dialogue.

FUTURE RESEARCH DIRECTIONS AND AUTOTUTOR EXTENSIONS

One important future direction is to improve the student modeling and conversational facilities of AutoTutor in order to minimize some of its persistent blemishes. This can be accomplished in a number of ways. There can be checks in the authoring tools to make sure that the content is complete when it is prepared by the author of the curriculum scripts. We have tried to correct this by developing facilities to improve the ease and quality of authoring the curriculum scripts, but this has not been an easy road. The best authors of content have some nontrivial expertise in information technologies, language, and discourse rather than being an instructor without such expertise. There needs to be a more intense research effort on understanding the process of authoring content in AutoTutor as well as other advanced learning environments.

A second direction is to develop systems that analyze language and discourse at deeper levels. Researchers can move beyond LSA and regular expressions and into more structure-sensitive processing and semantic decomposition (Olney, 2009; Olney, Graesser, & Person, 2010; Rus, McCarthy, McNamara, & Graesser, 2008). The dialogue manager module can move beyond lists of production rules and finite-state grammars and into the realm of recursive, complex planning and multiple-goal agendas. This approach of deeper natural language processing and discourse management is currently being pursued by Adrew Olney's Guru (see chapter XX by Olney, Person, & Graesser, 2010) in the area of biology and by Vasile Rus's DeepTutor in the area of physics. These extensions of AutoTutor are currently under development in the Institute for Intelligent Systems at the University of Memphis.

A third direction is to develop more sophisticated question answering facilities so that AutoTutor can answer a broad diversity of questions. This would contribute to mixed-initiative dialogue and put more control in the hands of the learner. There currently is a community of researchers who are exploring and testing computational models of question generation in learning environments (Rus & Graesser, 2009; see chapter xx). These efforts should contribute significantly to this direction.

A fourth direction is to build a version of AutoTutor that is sensitive to the learner's emotions and motivational states. D'Mello and Graesser (in press, chapoter xx) discusses this exciting extension. Related approaches to address motivation include: improvements in the agents speech and visual displays, implementing dimensions of personality in the agents, and integrating game-based facilities.

A fifth direction is to build a system that enhances AutoTutor's scalability. Xiangen Hu's AutoTutor-Lite is a minimalistic version of AutoTutor that includes the AutoTutor-style interface and interaction (animated agent and natural language conversation), but with a lightweight language analyzer and dialogue manager. AutoTutor-Lite has excellent authoring tools that lesson planners and instructors can use, even when they have minimal computer skills. Moreover, AutoTutor-Lite can be applied to powerpoint content on any verbal subject matter, is easily customizable, and can be integrated into e-learning environments on the web as well as the desktop. One can imagine an industry that "autotutorizes" the conventional eLearning content that is widely available.

Finally, AutoTutor has been a component in more comprehensive advanced learning environments. The Human Use Regulatory Affairs Advisor (HURAA) trains military personnel on research ethics in a web facility that has a full suite of learning modules, including an AutoTutor-like navigational guide (Hu & Graesser, 1994). Operation ARIES! (Millis, Cai, Graesser, Halpern, & Wallace, 2009) helps students learn about scientific methods in a game environment that includes an eBook with 22 chapters and case studies that are critiqued by students regarding scientific flaws. This system is guided by multiple animated agents, including a tutor agent and a peer agent; the human student is both tutored by the human agent and actively tutors the student agent. Danielle McNamara also has developed trainers with multiple interactive agents, as in the case of iSTART for teaching self explanations during reading (McNamara et al., 2004, 2007, see chapter xx) and of W-Pal for teaching writing (McNamara et al., chapter xx).

We believe that researchers have only begun to scratch the surface of using animated pedagogical agents with natural language interaction. Agents can have an endless number of dialogue styles, strategies, personalities, and physical features. We have developed one AutoTutor version that is emotionally supportive and another version that tries to shake up the emotions of the student by being rude and pretentiously telling the student what emotion the student is having. The rude AutoTutor is very engaging for some students whereas others would rather interact with the polite tutor. Student motivation may improve when the agents are matched to the cognitive, personality, emotional, and social profiles of individual students. The world of pedagogical agents is indeed on par with communication with other humans.

REFERENCES

Anderson, J. R. & Gluck, K. (2001). What role do cognitive architectures play in intelligent tutoring systems? In D. Klahr & S. M. Carver (Eds.) *Cognition & Instruction: Twenty-five years of progress*, 227-262. Hillsdale, NJ: Erlbaum.

Baylor, A. L., & Kim, Y. (2005). Simulating instructional roles through pedagogical agents. *International Journal of Artificial Intelligence in Education, 15*, 5–115.

Chi, M.T.H., Siler, S.A. & Jeong, H. (2004). Can tutors monitor students' understanding accurately? *Cognition and Instruction, 22*(3), 363-387.

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.

Corbett, A.T. (2001). Cognitive computer tutors: Solving the two-sigma problem. *User Modeling: Proceedings of the Eighth International Conference, UM 2001,* 137-147.

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

D'Mello, S., King, B., Chipman, P., & Graesser, A.C. (in press). Towards spoken human-computer tutorial dialogues. *Human Computer Interaction*.

D'Mello, S., Olney, A. M., & Person, N. (in press). Mining collaborative patterns in tutorial dialogues. *Journal of Educational Data Mining*.

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

Evens, M., Michael, J. (2006). *One-on-one tutoring by humans and machines*. Mahwah, NJ: Erlbaum.

Gholson, B., Witherspoon, A., Morgan, B., Brittingham, J. K., Coles, R., Graesser, A. C., Sullins, J., & Craig, S. D. (2009). Exploring the deep-level reasoning questions effect during vicarious learning among eighth to eleventh graders in the domains of computer literacy and Newtonian physics. *Instructional Science, 37*, 487-493.

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

Graesser, A. C., D'Mello, S., & Person, N. K. (2009). Metaknowledge in tutoring. In D. Hacker, J. Donlosky, & A. C. Graesser (Eds.), *Handbook of metacognition in education* (pp. 361-382). New York: Taylor & Francis.;

Graesser, A. C., Jeon, M., & Dufty, D. (2008). Agent technologies designed to facilitate interactive knowledge construction. *Discourse Processes*, *45*, 298–322.

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. *Behavior Research Methods*, *Instruments*, *and Computers*, *36*, 180–193.

Graesser, A. C., McNamara, D. S., & VanLehn, K. (2005). Scaffolding deep comprehension strategies through Point&Query, AutoTutor, and iSTART. *Educational Psychologist, 40*, 225-234.

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.

Graesser, A. C., Person, N., Harter, D., & the Tutoring Research Group. (2001). Teaching tactics and dialog in AutoTutor. *International Journal of Artificial Intelligence in Education*, *12*, 257–279.

Graesser, A. C., Person, N. K., & Magliano, J. P. (1995). Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology*, *9*, 495–522.

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.

Graesser, A.C., Moreno, K., Marineau, J., Adcock, A., Olney, A., & Person, N. (2003). AutoTutor improves deep learning of computer literacy: Is it the dialog or the talking head? In U. Hoppe, F. Verdejo, and J. Kay (Eds.), *Proceedings of Artificial Intelligence in Education*. (pp, 47-.54). Amsterdam: IOS Press.

Hestenes, D., Wells, M., and Swackhamer, G. (1992), Force concept inventory. *The PhysicsTeacher*, *30*, 141-158.

Hu, X., & Graesser, A. C. (2004). Human Use Regulatory Affairs Advisor (HURAA): Learning about research ethics with intelligent learning modules. *Behavior Research Methods*, *Instruments*, *and Computers*, *36*, 241–249.

Jackson, G. T., & Graesser, A. C. (2006). Applications of human tutorial dialog in AutoTutor: An intelligent tutoring system. *Revista Signos*, *39*, 31–48.

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., Olney, A., Graesser, A. C., & Kim, H. J. (2006). AutoTutor 3-D simulations: Analyzing user's actions and learning trends. In R. Son (Ed.), *Proceedings of the 28th Annual Meeting of the Cognitive Science Society* (pp. 1557–1562). Mahwah, NJ: Erlbaum.

Johnson, W.L. and Valente, A. (2008). Tactical Language and Culture Training Systems: Using Artificial Intelligence to Teach Foreign Languages and Cultures. In *Proceedings of the 20th Innovative Applications of Artificial Intelligence (IAAI) Conference*.

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.

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

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.

Litman, D.J, Rose, C.P., Forbes-Riley, K., VanLehn, K., Bhembe, D., and Silliman, S. (2006). Spoken versus typed human and computer dialogue tutoring. *International Journal of Artificial Intelligence in Education*, *16*, 145-170.

McNamara, D. S., Levinstein, I. B., & Boonthum, C. (2004). iSTART: Interactive strategy training for active reading and thinking. *Behavioral Research Methods, Instruments, and Computers, 36*, 222 - 233.

McNamara, D. S., O'Reilly, T., Rowe, M., Boonthum, C., & Levinstein, I. B. (2007). iSTART: A web-based tutor that teaches self-explanation and metacognitive reading strategies. In D.S. McNamara (Ed.), *Reading comprehension strategies: Theories, interventions, and technologies*. Mahwah, NJ: Erlbaum.

Millis, K., Cai, Z., Graesser, A., Halpern, D. & Wallace, P. (2009). Learning scientific inquiry by asking questions in an educational game. In T. Bastiaens et al. (Eds.), *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education* (pp. 2951-2956). Chesapeake, VA: AACE.

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

Moreno, R., & Mayer, R. E. (2007). Interactive multimodal learning environments. *Educational Psychology Review, 19*, 309-326.

Olney, A. M. (2009). GnuTutor: An open source intelligent tutoring system based on AutoTutor. In *Proceedings of the 2009 AAAI Fall Symposium on Cognitive and Metacognitive Educational Systems* (pp. 70-75). Washington, DC: AAAI Press.

Olney, A., Louwerse, M., Mathews, E., Marineau, J., Hite-Mitchell, H., & Graesser, A. (2003). Utterance classification in AutoTutor. In J. Burstein & C. Leacock (Eds.), *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing*. Philadelphia: Association for Computational Linguistics.

Olney, A. M., Graesser, A. C., & Person, N. K. (2010). Tutorial dialog in natural language. In R. Nkambou, R. Mizoguchi, & J. Bourdeau (Eds.), *Advances in Intelligent Tutoring Systems*. (pp. xx-xx). New York: Springer.

Olney, A. M., Person, N. K., & Graesser, A. C. (2010). Guru: Designing a conversational expert intelligent tutoring system. In *Applied natural language processing and content analysis: Identification, investigation, and resolution*. Amsterdam: IOS Press.

Person, N. K., Graesser, A. C., & the Tutoring Research Group. (2002, June). *Human or computer?: AutoTutor in a bystander Turing test*. Paper presented at the Intelligent Tutoring Systems 2002 Conference, Biarritz, France.

Pon-Barry, H., Clark, B., Schultz, K., Bratt, E O., Peters, S., & Haley, D. (2005) Contextualizing reflective dialogue in a spoken conversational tutor. *Educational Technology & Society*, *8*, 42-51.

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. (2006). Deeper natural language processing for evaluating student answers in intelligent tutoring systems. *Proceedings of the American Association of Artificial Intelligence*. Menlo Park, CA: AAAI.

Rus, V. & Graesser, A.C. (Eds.). (2009). *The question generation shared task and evaluation challenge.* ISBN: 978-0-615-27428-7.

Rus, V., McCarthy, P. M., McNamara, D. S., & Graesser, A. C. (2008). A study of textual entailment. *International Journal on Artificial Intelligence Tools*, *17*, 659–685.

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

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., Jordan, P., Rosé, C. P., et al. (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.

Woolf, B. (2009). *Building Intelligent Interactive Tutors: Student-Centered Strategies for Revolutionizing E-learning*, San Francisco, CA: Elsevier Inc., Morgan Kauffman.

ADDITIONAL READING SECTION

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., Corbett, A.T., Koedinger, K.R., & Wagner, A.Z. (2004). Off-task behavior in the cognitive tutor classroom: When students "Game the System". *Proceedings of ACM CHI 2004: Computer-Human Interaction*, 383-390.

Biswas, G., Leelawong, K., Schwartz, D., Vye, N. & The Teachable Agents Group at Vanderbilt (2005). Learning by teaching: A new agent paradigm for educational software. *Applied Artificial Intelligence*, *19*, 363-392.

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

Bull, S. & Kay, J. (2007). Student models that invite the learner in: The SMILI open learner modeling framework. *International Journal of Artificial Intelligence in Education 17*, 89-120.

Chi, M. T. H., Roy, M., & Hausmann, R. G. M. (2008) Observing tutorial dialogues collaboratively: Insights about human tutoring effectiveness from vicarious learning. *Cognitive Science*, *32*(2), 301-341.

Chi, M.T.H., Siler, S., Yamauchi, T., Jeong, H. & Hausmann, R. (2001). Learning from human tutoring. *Cognitive Science*, *25*, 471- 534.

Cole, R., van Vuuren, S., Pellom, B., Hacioglu, K., Ma, J., & Movellan, J., (2003). Perceptive animated interfaces: First steps toward a new paradigm for human computer interaction. *Proceedings of the IEEE*, *91*, 1391-1405.

Collins, A., & Halverson, R. (2009). *Rethinking education in the age of technology: The digital revolution and schooling in America.* New York: Teacher College Press.

Corbett, A. T., & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling & User-Adapted Interaction*, *4,* 253-278.

Craig, S. D., Sullins, J., Witherspoon, A. & Gholson, B. (2006). Deep-level reasoning questions effect: The role of dialog and deep-level reasoning questions during vicarious learning. *Cognition and Instruction, 24(4),* 563-589.

D'Mello, S.K., Picard, R., & Graesser, A.C. (2007). Toward an affect-sensitive AutoTutor. *IEEE Intelligent Systems, 22*, 53-61.

Fuchs, L., Fuchs, D., Bentz, J., Phillips, N., & Hamlett, C. (1994). The nature of students' interactions during peer tutoring with and without prior training and experience. *American Educational Research Journal, 31*, 75-103.

Gholson, B., Witherspoon, A., Morgan, B., Brittingham, J. K., Coles, R., Graesser, A. C., Sullins, J., & Craig, S. D. (2009). Exploring the deep-level reasoning questions effect during vicarious learning among eighth to eleventh graders in the domains of computer literacy and Newtonian physics. *Instructional Science, 37*, 487-493.

Graesser, A. C., Conley, M.W., & Olney, A. (in preparation). Intelligent tutoring systems. In S. Graham and K. Harris (Eds.), *APA Handbook of Educational Psychology.* Washington, DC: American Psychological Association.

Gratch, J., Rickel, J., Andre, E., Cassell, J., Petajan, E., & Badler, N. (2002). Creating interactive virtual humans: Some assembly required. *IEEE Intelligent Systems, 17*, 54-63.

Kolodner, J., Cox, M., & Gonzalez-Calero, P. (2005). Case-based reasoning-inspired approaches to education. *The Knowledge Engineering Review*, *20*(3), 299-303.

Lepper, M. R., & Woolverton, M. (2002). The wisdom of practice: Lessons learned from the study of highly effective tutors. In J. Aronson (Ed.), *Improving academic achievement: Impact of psychological factors on education* (pp. 135-158). Orlando, FL: Academic Press.

McQuiggan, S., & Lester, J. (2007). Modeling and evaluating empathy in embodied companion agents. *International Journal of Human-Computer Studies*, 65, 348-360.

Moreno, R., & Mayer, R. E. (2004). Personalized messages that promote science learning in virtual environments. *Journal of Educational Psychology, 96*, 165-173.

Palincsar, A.S., & Brown, A.L. (1984). Reciprocal teaching of comprehension- fostering and monitoring activities. *Cognition and Instruction*, *1*, 117-175.

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

Reeves, B. and Nass, C. (1996). *The Media Equasion: how people treat computers, televisions, and new media like real people and places*. University Press, Stanford, California.

Roscoe, R.D., & Chi, M.T.H. (2007). Understanding tutor learning: Knowledge-building and knowledge-telling in peer tutors' explanations and questions. *Review of Educational Research, 77*, 534-574.

Schwartz, D.L., & Bransford, J.D. (1998). A time for telling. *Cognition & Instruction, 16(4),* 475-522.

KEY TERMS & DEFINITIONS

Agent. A computer module that intelligently interacts with a human by detecting states of the human and system and by responding to the human in a fashion that achieves specific goals.

Authoring tool. A computer facility for creating and modifying content in a computerized learning environment

Curriculum script. The subject matter knowledge that is stored in the computer on the topic being tutored. The learning management system accesses and uses this content during the course of producing responses to the student.

Dialogue management. The algorithms that the system uses to track the student's knowledge, the dialogue history, and the state of the system, and to respond in a fashion that achieves conversational goals in a coherent manner.

Latent semantic analysis (LSA). A statistical representation of words and world knowledge that is based on the words that are used in a large corpus of documents.

Metacognition. Knowledge about nature of cognitive states and processes.

Pattern matching and pattern completion. Pattern matching is an algorithm that computes the extent to which student contributions match expectations. Pattern completion is the generation of actions that attempt to achieve high pattern matches.

Semantic matching. A comparison between two texts on the degree to which they have similar meaning. For example, student expressions are compared with expectations of the tutor.

Speech act. A statement or utterance in a conversational turn that has a particular discourse function, such as a question, command, assertion, promise, or feedback expression.

Student modeling. The computer algorithms for tracking what the student knows about subject matter knowledge.