International Journal of Artificial Intelligence in Education (2001), to appear

Teaching Tactics and Dialog in AutoTutor

Arthur C. Graesser, Natalie K. Person. Derek Harter and The Tutoring Research Group

Arthur Graesser, Department of Psychology, 202 Psychology Building, The University of Memphis, Memphis, TN 38152-3230, a-graesser@memphis.edu
http://mnemosyne.csl.psyc.memphis.edu/home/graesser/

Natalie K. Person, Rhodes College, Department of Psychology, 2000 North Parkway, Memphis, TN 38112 person@rhodes.edu

Derek Harter, Department of Mathematical Sciences, he University of Memphis, Memphis, TN 38152, dharter@memphis.edu

Abstract. The Tutoring Research Group at the University of Memphis has developed a computer tutor (called AutoTutor) that simulates the discourse patterns and pedagogical strategies of a typical human tutor. The dialog tactics were based on a previous project that dissected 100 hours of naturalistic tutoring sessions. AutoTutor is currently targeted for college students in introductory computer literacy courses, who learn the fundamentals of hardware, operating systems, and the Internet. Instead of merely being an information delivery system, AutoTutor serves as a discourse prosthesis (or collaborative scaffold) that assists the student in actively constructing knowledge. A dialog manager coordinates the conversation that occurs between a learner and a pedagogical agent, whereas lesson content and world knowledge are represented in a curriculum script and latent semantic analysis. The agent is a talking head with discourse-sensitive facial expressions and synthesized speech. Evaluations of AutoTutor have shown that the tutoring system improves learning and memory of the lessons by .5 to .6 standard deviation units. This article describes the components of AutoTutor and contrasts two versions that follow somewhat different teaching tactics.

INTRODUCTION

It is widely acknowledged in the field of education that students rarely acquire a deep understanding of the material they are supposed to learn in their courses. Students normally settle for shallow knowledge, such as lists of concepts, a handful of facts about each concept, and simple definitions of key terms. Students lack the deep coherent explanations that organize the shallow knowledge and that fortify the learner for generating inferences, solving problems, and applying their knowledge to practical situations. They lack the skill of articulating and manipulating symbols, formal expressions, and precise quantities. They lack the ability to forecast how a complex system will behave when given different inputs. The acquisition of shallow knowledge is unfortunately reinforced by the traditional classroom activities and testing formats. Most classroom lectures are information delivery systems for shallow knowledge. Most teachers' questions are short-answer questions that require only single words or short phrases in the student responses. The format of most examinations consists of multiple choice, true-false, or fill-in-the-blank questions that, once again, tap primarily the shallow knowledge. Given this unfortunate state of affairs, many researchers and teachers in education have been exploring learning environments and pedagogical strategies that promote deep comprehension.

Constructivism is the most popular general approach to cracking the barrier of shallow knowledge (Biggs, 1996; Bransford, Goldman, & Vye, 1991; Brown, 1988; Chi, deLeeuw, Chiu, & LaVancher, 1994; Palincsar & Brown, 1984; Papert, 1980; Piaget, 1952; Pressley & Wharton-McDonald, 1997; Rogoff, 1990; VanLehn, Jones, & Chi, 1992; Vygotsky, 1978). According to this approach, the learner needs to actively construct meanings and knowledge by interacting with the world and other people. Moreover, the premier meaning representations are coherent, explanatory, and deeply rooted in the learner's experiential knowledge base. Learning environments should stimulate active construction of knowledge and provide feedback on these constructions rather than being mere information delivery systems. *Dialectical constructivism* stipulates that complex learning primarily occurs through an interaction between learners and their environments, whereas *exogenous constructivism* emphasizes the constraints of the outside world and *endogenous constructivism* emphasizes the cognitive and biological constraints of the learner (Moshman, 1982). Constructivist approaches have been so compelling that they have shaped the standards for curriculum and instruction in the United States during the last decade, e.g., *Standards for the English Language Arts* (NCTE, 1996), *Curriculum and Evaluation Standards for School Mathematics* (NCTM, 1989), *National Science Education Standards* (NRC, 1996).

We believe that one-on-one tutoring is an ideal learning environment for investigating the constructive processes systematically. There are only two agents, namely the learner and the student, so it is possible to systematically manipulate tutoring tactics and to observe the impact on learning gains. It is an open question of whether the tutor should be a human or a computer. There seem to be advantages and limitations of both types of agent. For example, human tutors are presumably more personable, responsive, and deeper. But they also get tired, are sometimes irritated, and are not always consistent in implementing pedagogical strategies. A new wave of computer tutors are also simulating some of these human qualities, so maybe computer tutors will be viable alternatives. One such computer tutor (called AutoTutor) will be the focus of this article.

THE EFFECTIVENESS OF ONE-ON-ONE TUTORING

At the heart of AutoTutor is the assumption that one-on-one tutoring is a powerful method of promoting knowledge construction. In fact, there is substantial empirical evidence that human tutoring is extremely effective when compared to typical classroom environments. Cohen, Kulik, and Kulik (1982) performed a meta-analysis on a large sample of studies that compared human-to-human tutoring with classroom controls. The vast majority of the tutors in these studies were untrained in tutoring skills and had moderate domain knowledge; they were peer tutors, cross-age tutors, or paraprofessionals, but rarely accomplished professionals. These "unaccomplished" human tutors enhanced learning with an effect size of .4 standard deviation units, which translates to approximately a half a letter grade. Accomplished human tutors do substantially better according to Bloom (1984), who reported an effect side of 2.0 standard deviation units in learning gains (or 2 letter grades approximately). So the advantage of human-to-human tutoring over the classroom appears to vary between .4 and 2.0 standard deviation units, depending on the expertise of the tutor.

Computer tutors have implemented some of the pedagogical strategies that have been advocated in education. One advantage of computer tutors is that particular pedagogical strategies can be manipulated, as opposed to merely observed naturalistically. This makes it easier to determine whether a particular pedagogical component has a causal impact on learning gains. Moreover, the computer tutors have proven to be effective. AutoTutor implemented the pedagogical strategies and dialogue patterns of unaccomplished tutors and produced learning gains of .5 to .6 standard deviation units (Graesser, Bautista, Link, Kreuz, & TRG, 2001; Graesser, VanLehn, Rose, Jordan, Harter, in press; Link, Pomeroy, DiPaolo, Rajan, Klettke, Bautista, Kreuz, Graesser, & TRG, 2000). During the last 20 years, intelligent tutoring systems (ITS) have implemented several systematic strategies for promoting learning, such as the error identification and correction, building on prerequisites, frontier learning (expanding on what the learner already knows), student modeling (inferring what the student knows and having that guide tutoring strategies), and building coherent explanations (Anderson, Corbett, Koedinger, & Pelletier, 1995; Gertner, & VanLehn, 2000; Koedinger, Anderson, Hadley, & Mark, 1997; Lesgold, Lajoie, Bunzo, & Eggan, 1992; Sleeman & Brown, 1982; vanLehn, 1990). The ITSs that have been successfully implemented and tested (such as VanLehn's Andes physics tutor and Koedinger's PACT algebra tutor) have produced learning gains of approximately 1.0 standard deviation unit, i.e., one letter grade (Corbett, Anderson, Graesser, Koedinger, & VanLehn, 1999; du Boulay, 2000). According to the available evidence, the learning gains of the sophisticated ITSs (1.0 SD) are higher than those of unaccomplished human tutors (.4 SD), but not quite as good as the accomplished human tutors (2.0 SD). AutoTutor's performance (.5 to .6 SD) is on par with unaccomplished human tutors.

It appears that there are two different mechanisms that potentially explain the effectiveness of tutoring (Corbett et al., 1999; Graesser, Person & Magliano, 1995). The first is the sophisticated tutoring strategies that have been identified in the ITS literature. The second is the dialog patterns and natural language that help the tutor scaffold the learner to new levels of mastery. According to Graesser et al. (1995) and the theoretical foundation of AutoTutor, there is something about discourse and natural language (as opposed to sophisticated pedagogical strategies) that explains the effectiveness of unaccomplished human computers. The performance assessments of ITS systems indicate that the sophisticated tutoring strategies move the tutoring process one giant step further. Therefore, the ideal computer tutor would presumably embrace both of these mechanisms.

WHAT DO UNACCOMPLISHED TUTORS DO?

AutoTutor incorporate features of tutorial dialog that are prevalent in normal tutoring sessions with unaccomplished human tutors. In previous research projects funded by the Office of Naval Research, Graesser and Person videotaped, transcribed, and analyzed nearly 100 hours of naturalistic tutoring sessions (Graesser & Person, 1994; Graesser et al., 1995; Person & Graesser, 1999). The corpus of tutoring sessions included (a) graduate students tutoring undergraduates on the fundamentals of research methods and (b) middle school students tutoring younger students in basic algebra. After analyzing this rich corpus, Graesser and Person discovered what tutors do versus do not do during most tutoring sessions. Our discoveries were enlightening and often counterintuitive.

The question arises as to why we analyzed and simulated an unaccomplished tutor rather than a skilled professional tutor. There are several reasons. First, transcripts were not available for a reasonable sample of tutors who had both tutoring training and high expertise in the subject matter. Thus, accomplished tutoring expertise was unavailable, as is typically the case in nearly all school settings. Second, we were convinced that it was technologically feasible to simulate the discourse and pedagogical strategies of unaccomplished tutors, but not accomplished tutors. Third, unaccomplished tutors are known to be effective in producing learning gains, so simulating such mechanisms is worthwhile.

Our anatomy of normal tutoring sessions revealed that the tutors do not use most of the ideal tutoring strategies that have been identified in education and the ITS enterprise. These strategies include the Socratic method (Collins, 1985), modeling-scaffolding-fading (Collins, Brown, &

Newman, 1989; Rogoff, 1990), reciprocal training (Palincsar & Brown, 1984), anchored situated learning (Bransford et al., 1991), error diagnosis and remediation (Sleeman & Brown, 1982), frontier learning, building on prerequisites (Gagne, 1977), and sophisticated motivational techniques (Lepper, Woolverton, Mumme, & Gurtner, 1991). Detailed discourse analyses have been performed on small samples of accomplished tutors in an attempt to identify sophisticated tutoring strategies (Fox, 1993; Hume, Michael, Rovick, & Evens, 1996; Merrill, Reiser, Ranney, & Trafton, 1992; Moore, 1995; Putnam, 1987). However, we discovered that nearly all of these sophisticated tutoring strategies were virtually nonexistent in the naturalistic tutoring sessions that we videotaped and analyzed (Graesser et al., 1995; Person & Graesser, 1999). Tutors clearly need to be trained how to use the sophisticated tutoring tactics because they do not routinely emerge in typical tutoring sessions with untrained tutors.

The 5-step dialog frame is one of the prominent dialog patterns in naturalistic tutoring (Graesser & Person, 1994). The five steps in this frame are presented below.

Step 1: Tutor asks question (or presents problem)
Step 2: Learner answers question (or begins to solve problem)
Step 3: Tutor gives short immediate feedback on the quality of the answer (or solution)
Step 4: The tutor and learner collaboratively improve the quality of the answer.
Step 5: The tutor assesses the learner's understanding of the answer

This 5-step frame has been adopted in AutoTutor. This 5-step dialog frame in tutoring is a significant augmentation over the 3-step pattern that is prevalent in classroom instruction. That is, Mehan (1979) and others have reported a 3-step pattern that is often called IRE: Initiation (a question or claim articulated by the teacher), Response (an answer or comment provided by the student) and Evaluation (the teacher evaluates the student contribution). These IRE steps directly correspond to steps 1, 2, and 3 of the 5-step dialog frame for tutoring. Graesser et al. (1995) argued that the advantage of tutoring over the classroom lies primarily the lengthy multi-turn exchange in step 4. Another possibility might be Step 5. However, our anatomy of naturalistic tutoring revealed that tutors only minimally assess the learner's understanding of the student in step 5. The tutor normally asks "Do you understand?" and then the vast majority of student responses are positive ("Yes"), even though most of the students have a vague, incomplete, or incorrect understanding (Person, Graesser, Magliano, & Kreuz, 1994). In fact, it is the better students who tend to answer "No" to these comprehension-gauging questions, perhaps because they are more self-regulated learners or have more fine-tuned metacognitive strategies (Hacker, Dunlosky, & Graesser, 1998). An ideal tutor would press the student further by asking follow-up questions that diagnose whether the student truly understands the answer.

Our anatomy of human tutoring revealed that human tutors generate dialog moves that are sensitive to the quality and quantity of the preceding student turn. They adapt to what the student says, but do not analyze the student's knowledge at a fine-grained level. The tutor presents dialog moves that are both responsive to the student and that facilitate the student in actively constructing knowledge. The tutor dialog move categories that we identified in human tutoring sessions are provided below.

(1) Positive immediate feedback. "That's right" "Yeah"
(2) Neutral immediate feedback. "Okay" "Uh-huh"
(3) Negative immediate feedback. "Not quite" "No"
(4) Pumping for more information. "Uh-huh" "What else"
(5) Prompting for specific information. "The primary memories of the CPU are ROM and _____"
(6) Hinting. "What about the hard disk?"

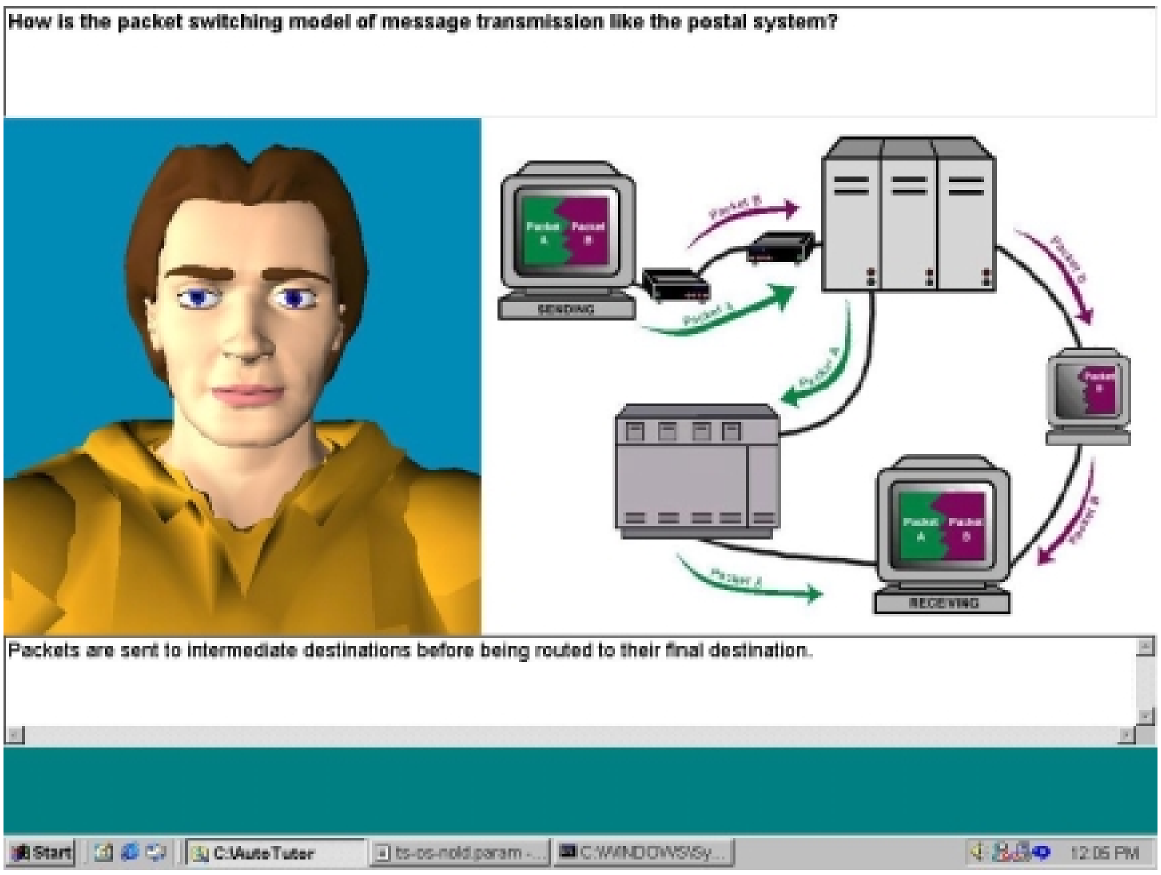
(7) Elaborating. “CD ROM is another storage medium.”
(8) Splicing in the correct content after a student error.
(9) Summarizing. "So to recap," <succinct recap of answer to question>

After spending nearly a decade toiling over thousands of pages of tutoring transcripts and staring at hundreds of hours of videotaped tutoring sessions, we decided to it was time to put our knowledge to good use and build AutoTutor.

WHAT IS AUTOTUTOR?

The Tutoring Research Group (TRG) at the University of Memphis has been developing a computer tutor, called AutoTutor, that simulates a typical human tutor (Graesser, Franklin, Wiemer-Hastings, & TRG, 1998; Graesser, VanLehn, et al., in press; Graesser, Wiemer-Hastings, Wiemer-Hastings, Kreuz, & TRG, 1999; Person, Graesser, Kreuz, & Pomeroy, & TRG, in press; Wiemer-Hastings, Graesser, Harter, & TRG, 1998). AutoTutor attempts to comprehend student contributions and to simulate dialog moves of human tutors. AutoTutor-1 attempts to simulate the dialog moves of normal (unskilled) tutors, whereas AutoTutor-2 incorporates more sophisticated tutoring strategies. AutoTutor is currently being developed for college students who take an introductory course in computer literacy. These students learn the fundamentals of computer hardware, the operating system, and the Internet. AutoTutor is written in the Java programming language and is currently implemented on Pentium Computers in an NT operating system.

A brief snapshot of AutoTutor-1 in action should concretize the nature of AutoTutor. AutoTutor works by having a conversation with the learner. AutoTutor appears as a talking head that acts as a dialog partner with the learner. The talking head delivers AutoTutor’s dialog moves with synthesized speech, intonation, facial expressions, and gestures. The major question (or problem) that the learner is working on is both spoken by AutoTutor and is printed at the top of the screen (see Figure 1). The major questions are generated systematically from a curriculum script, a module that will be discussed later. AutoTutor’s major questions are *not* the fill-in-the blank, true/false, or multiple-choice questions that are so popular in the US educational system. Instead, the questions invite lengthy explanations and deep reasoning (e.g., answers to *why, how, what-if* questions). The goal is to encourage students to articulate lengthier answers that exhibit deep reasoning, rather than to recite short snippets of shallow knowledge. There is a continuous multi-turn tutorial dialog between AutoTutor and the learner during the course of answering a major question (or solving a problem). When considering both the learner and AutoTutor, it typically takes 10 to 30 turns during the tutorial dialog when a single question from the curriculum script is answered. The learner types in his/her contributions during the exchange by keyboard. For some topics, there are graphical displays and animation, with components that AutoTutor points to. AutoTutor was designed to be a good conversation partner that comprehends, speaks, points, and displays emotions, all in a coordinated fashion.



How is the packet switching model of message transmission like the postal system?

Packets are sent to intermediate destinations before being routed to their final destination.

Figure 1: AutoTutor with Marco Talking Head

The exchange below is one of the shorter tutorial dialogs between a college student and AutoTutor. This student was an above average student in the computer literacy class.

01 AUTOTUTOR: When you turn on the computer, how is the operating system first activated and loaded into RAM?
02 STUDENT: When the computer is turned on a file is automatically booted up.
03 AUTOTUTOR: Anything else?
04 STUDENT: This file contains the operating system.
05 AUTOTUTOR: What else?
06 STUDENT: The operating system is then read into RAM and activated.
07 AUTOTUTOR: Right. The CPU executes instructions in ROM, which helps the operating system to start.
08 STUDENT: After being read from ROM the OS is activated.
09 AUTOTUTOR: Right. Let’s review what we’ve just gone over. The operating system must first be stored on the hard disk. When you turn on the computer, the CPU executes instructions stored in ROM. These instructions help boot the operating system. The operating system is then loaded from the hard disk into RAM.

AutoTutor first asks a *how*-question in turn 1. Then the student provides answer information in turns 2, 4, 6, and 8. These answers are encouraged by AutoTutor’s *pumps* in turns 3 and 5 (i.e., “Anything else?”, “What else?”). AutoTutor delivers these pumps in an effort to encourage the student to actively construct knowledge. Thus, instead of being an *information delivery system* that bombards the student with a large volume of information, AutoTutor is a *discourse prosthesis* that attempts to get the student to do the talking and that explores what the student knows. As discussed earlier, AutoTutor adopts a constructivist theoretical view that a key feature of effective learning lies in assisting students in actively constructing subjective explanations and elaborations of the material, as students answer questions and solve problems that require deep reasoning. At the same time, however, the students need to answer enough questions and to solve enough problems for them to understand the constraints of the domain knowledge. It would not be good for the student to flounder unproductively for a long time. Consequently, AutoTutor sometimes needs to bring the student back on track by supplying cues and clues that lead to the evolution of a complete answer to the question. These clues include *hints, prompts* for the student to fill in a word or phrase, and *assertions* that fill in missing ideas. The student had forgotten about the role of ROM in launching the operating system, so AutoTutor brings up ROM in turn 7. The student builds on this suggestion in turn 8. At that point, the important pieces of a good complete answer have been covered, so AutoTutor summarizes the answer in turn 9. AutoTutor periodically gives positive immediate *feedback* after the student contributions (i.e., “right.”). This feedback is not only motivating, but creates the impression that AutoTutor is listening to what the student is communicating. These characteristics of a tutorial exchange are quite similar to discourse patterns in normal tutoring between humans.

It is important to point out that there is an asymmetry in the communication media between AutoTutor and the learner. AutoTutor talks, points, and displays information whereas the only input channel for the student is typing information via the keyboard. It presumably would be better for both conversational participants to speak, point, gesture, and display information. It certainly would be more natural. The Tutoring Research Group is indeed exploring this alternative. However, there are widely acknowledged technical challenges in building a reliable speech recognition system for continuous speech. The technologies that interpret user’s gestures, head nods, facial expressions, and graphical input are even more undeveloped. When these technologies are more available and reliable, they can be explored, tested, and compared. At present, however, AutoTutor is confined to keyboard input from the user. It could perhaps be argued that keyboard input from the learner may end up producing higher learning gains than the evanescent speech input. It forces the user to articulate their answers in writing, more carefully thought out, with language that is more technical than conversational. The impact of alternative communication media on learning gains is one important direction for future research (see Whittaker, in press).

MECHANISMS OF AUTOTUTOR

It is beyond the scope of this section to review all of the components of AutoTutor. Instead, the focus will be on those mechanisms that are most relevant to tutorial dialog and teaching tactics. At this point we have developed two versions of AutoTutor, which we refer to as *AutoTutor-1* and *AutoTutor-2*. AutoTutor-2 has somewhat different conversation parameters and dialog tactics that were added to get the student construct more information and do more of the talking in the mixed-initiative dialog.

Curriculum scripts with example problems, deep questions, graphics, and animation.

A curriculum script is a loosely ordered set of skills, concepts, example problems, and question-answer units. Most human tutors follow a script-like macrostructure, but briefly deviate from the structure when the student manifests difficulties, misconceptions, and errors. The content of the curriculum script in tutoring (compared with classrooms) has more deep reasoning questions (e.g., *why, how, what-if, what-if-not*), more problems to solve, and more examples (Graesser et al., 1995).

The curriculum script in AutoTutor organizes the topics and content of the tutorial dialog. The script includes didactic descriptions, tutor-posed questions, example problems, figures, and diagrams (along with anticipated good responses to each topic). There also is a glossary of technical terms with definitions (i.e., answers to the learner's "What does X mean?" questions). There were 36 topics (i.e., example problems or deep reasoning questions) in AutoTutor-1, 12 each for the hardware, the operating system, and Internet. Each topic is represented simply as a set words, sentences, or paragraphs in a free text format. Thus, it is easy for a lesson planner to create new topics and content with a simple authoring tool; there is no need to craft the content in structured LISP or Prolog code, which is routinely done when systems are created in most ITSs.

Associated with each topic is a focal question, a set of basic noun-like concepts, a set of ideal good answer aspects (each being roughly a sentence of 10-20 words), different forms of expressing or eliciting each ideal answer aspect (i.e., a hint, prompt, versus assertion), a set of anticipated bad answers (i.e., bugs, misconceptions), a correction for each bad answer, and a summary of the answer or solution. Except for the hints, prompts, and corrections, the preparation of curriculum script requires no special computer knowledge on the part of the lesson planner. The system was designed this way so that AutoTutor could be used for a large range of topics (virtually any topic except those that require the precision of mathematics) and so lesson planners could develop the content with minimal knowledge of discourse or computer science.

Verbal content

Appendix A shows a portion of one of the curriculum scripts on the topic of operating systems. This curriculum script was developed for AutoTutor-2. The difficulty level of this question is moderate, as opposed to easy or difficult. Context information is presented prior to the focal question: *How does the operating system of a typical computer process !several jobs simultaneously, with only !one CPU?* The explanation point symbol (!) is part of the mark-up language to the talking head; it designates that the word should be stressed in the synthesized speech. The ideal answer to this question is a lengthy answer that has 5 good answer aspects (i.e., expected good answers), signified as A_1, A_2, … A_5. Appendix A includes information about two of the 5 good answer aspects, but not the other three good answer aspects. For example, the first good answer aspect is: *The operating system helps the computer to work on several jobs simultaneously by rapidly switching back and forth between jobs*. There is a verbal description associated with each good answer aspect (preceded by pgood in Appendix A). The student's input, entered by keyboard, is constantly compared to each of the 5 good answer aspects as the dialog evolves, turn by turn. Latent Semantic Analysis (LSA), a component that will be discussed later, is used to assess the match between student contributions and each good answer aspect.

Very often, a student cannot articulate a particular good answer aspect. AutoTutor needs to provide some scaffolding to extract this information from the student or to get the student to articulate this knowledge. Assertions, hints, and prompts provide this scaffolding. Assertions succinctly articulate the desired information (signified by passert). Hints are questions (signified by phint) that lead the student to the desired information (signified by phintc). Prompts are assertions that that leave out the last word (signified by pprompt); when these get articulated, there is a stress intonation contour that strongly encourages or signals the student to fill in the desired word (signified by ppromptk). After the student types in the response to the prompt, AutoTutor presents the correct missing word or expressions in a prompt response (signified by pprompte). It should be noted that there are several prompts and hints for each good answer aspect. AutoTutor attempts to extract the desired information from the student by many different cues and clues.

The curriculum script also has a number of bad answers that capture common bugs and misconceptions. When a student expresses a contribution that matches one of the bad answers (signified by bad and bbad), then AutoTutor corrects the error by splicing in a correction (signified by splice).

Graphics and animation

AutoTutor has graphic displays and a small number of animation clips that coordinate graphic and animated displays, speech synthesis, facial expressions, communicative gestures, and pointing. Extensive use of multimedia can have its advantages. But there are also liabilities from "feature bloat" to the extent that a multimedia show splits the attention of the student (Sweller & Chandler, 1994). The coordination of the visual and auditory information was guided by recent research on multimedia, education, and cognitive science to the extent that research is available. For example, narrative information (i.e., what AutoTutor says) needs to be sequenced simultaneously with visual information (Mayer, 1997), text and pictures must be in spatial and temporal contiguity (Moreno & Mayer, 1999), and it is not a good policy to present lengthier narrative messages in both a text and an auditory modality simultaneously (Kalyuga, Chandler, & Sweller, 1999).

Natural language extraction and speech act classification

AutoTutor needs to classify the speech acts of student contributions in order to flexibly respond to what the student types in. AutoTutor segments the categorized string of words and punctuation marks within a learner's turn into speech act units, relying on punctuation to perform this segmentation. Then each speech act is assigned to one of the following speech act categories: Assertion, WH-question, YES/NO question, Metacognitive comments (*I don't understand*), Metacommunicative acts (*Could you repeat that?*), and Short Response. Speech act classification is performed by accessing lexicons, identifying the parts-of-speech of words, executing finite state transducers, and consulting frozen expression catalogues. Marineau et al. (2000) tested and compared speech act classifiers that had architectures with neural networks, syntactic parsers, versus surface feature extraction.

Latent Semantic Analysis

The fact that world knowledge is inextricably bound to the process of comprehending language and discourse is widely acknowledged, but researchers in computational linguistics and artificial intelligence have not had a satisfactory approach to handling the deep abyss of world knowledge. AI researchers have traditionally relied on semantic networks and conceptual graph structures (Lehmann, 1992; Lenat, 1995), which unfortunately take years to develop in extensive knowledge engineering efforts. Recently, Latent semantic analysis (LSA) has recently been proposed as a statistical representation of a large body of world knowledge (Kintsch, 1998; Landauer & Dumais, 1997; Landauer, Foltz, & Latham, 1998). An LSA space can be created and tested for a very large corpus of documents in a short period of time (less than a month). LSA provides the foundation for grading essays, even essays that are not well formed grammatically, semantically, and rhetorically; LSA-based essay graders can assign grades to assays as reliably as experts in composition (Foltz, 1996; Landauer et al., 1998). An LSA space is created after processing a large corpus of texts that are relevant to the topic being tutored. The LSA uses singular value decomposition to reduce a large Word by Document co-occurrence matrix to approximately 100-500 dimensions. LSA capitalizes on the fact that particular words appear in particular texts (called "documents"). Each word, sentence, or text ends up being a weighted vector on the K dimensions. The "match**"** (i.e., similarity in meaning, conceptual relatedness) between two words, sentences, or texts is computed as a geometric cosine (or dot product) between the two vectors, with values ranging from 0 to 1. The match between two language strings can be high even though there are few if any words in common between the two strings. LSA goes well beyond simple string matches because the meaning of a language string is partly determined by the company (other words) that each word keeps.

As mentioned earlier, AutoTutor successfully used LSA as the backbone for representing computer literacy. The quality of student answers were successfully computed from the assertions expressed during the student's turns. A student with high ability had (a) a high mean LSA between the assertions and good answer aspects and (b) a low mean LSA match with bad answers. LSA can evaluate the quality of the learner's answers as well graduate student research assistants in a computer literacy class (Graesser, Wiemer-Hastings, Wiemer-Hastings, Person, Harter, & TRG, 2000; Wiemer-Hastings, Wiemer-Hastings, Graesser, & TRG, 1999).

LSA does not have the capability of comprehending text at a deep level. It is essentially a knowledge-based, statistical pattern matcher. It does an impressive job evaluating the matches between student contributions and expected good answers or bad answers. However, LSA is not equipped to account for the order of words, syntax, logical expressions, quantification, negations, rhetorical relations between clauses, and other analytical components of comprehension. More traditional symbolic architectures in artificial intelligence and computational linguistics are needed to perform these analytical processes. In fact, we are currently developing hybrids between LSA and symbolic systems in our current versions of AutoTutor (Graesser, VanLehn, et al., in press). However, it should be recognized that the current symbolic systems face serious challenges handling scruffy, conversational discourse, so we would not be surprised if the analytical mechanisms accommodate a low percentage of user contributions. LSA is available when the analytical symbolic processors fail. Perhaps humans follow a similar processing trajectory.

Dialog management

Smooth conversation requires dialog management. There needs to be discourse markers and other cues that guide the student in the exchange and that can accommodate virtually any input of the student (Core, Moore, & Zinn, 2000; Freedman, 1999; Moore, 1995; Soller, Linton, Goodman, & Lesgold, 1999). A collaborative exchange between AutoTutor and the learner requires a mutual understanding of the turn-taking process. In human-to-human conversations, speakers signal to listeners that they are relinquishing the floor and that it is the listener's turn to say something (Clark, 1996; Nofsinger, 1991; Sacks, Schegloff, & Jefferson, 1978). However, human-to-computer conversations lack many of the subtle signals inherent to human conversations. When conversational agents lack turn-taking signals, the learner does not know when or if the learner is supposed to respond, and is sometimes confused when the tutor generates particular dialog moves. These problems have been minimized in AutoTutor after a systematic analysis of human tutorial dialog, the human-computer interface of AutoTutor, and rounds of testing.

Dialog management in AutoTutor has four subcomponents: (1) A Dialog Advancer Network (called the DAN), (2) a set of fuzzy production rules, (3) the selection of the next good answer aspect to cover, and (4) the articulation of a selected good answer aspect. These are the components that are primarily responsible for making AutoTutor an effective dialog partner so it is important to describe them in sufficient detail.

Dialog Advancer Network (DAN)

A dialog advancer network (DAN) manages the exchange by specifying appropriate discourse markers (e.g.,*Moving on, Okay*), dialog move categories, and frozen expressions within the tutor's turn. It does this in a fashion that is sensitive to the learner's previous turn. There are the following different categories of dialog moves that AutoTutor generates: main question, short feedback (i.e., positive, neutral, negative), pumps (*uh huh, tell me more*), prompts (*The primary memories of the CPU are ROM and _____*), prompt response *(and RAM*), hints, assertions (which is synonymous with "elaboration" in the remainder of this article), corrections, and summaries. The DAN is formally an augmented state transition network because the selection of a dialog move category on tutor turn N+1 is sensitive to a large space of parameters computed from the dialog history. The DAN in AutoTutor-1 does a fairly impressive job in managing the conversation, based on our performance data (Person, Graesser, & TRG).

Figure 2 is the DAN for AutoTutor-1. AutoTutor generates one or more dialog acts during turn N+1. The process begins by determining the primary speech act category in student turn N. This normally is easy to determine because the vast majority of student turns have only 1 or 2 speech acts. When there are more than one student speech acts, the most recent speech act is typically the one that is counted. AutoTutor then generates one or more dialog acts that are responsive to the student's contribution. If the student produces a frozen expression that requests the tutor to repeat itself (*Could you say that again?*), AutoTutor-1 produces a discourse marker (*Once Again*) and repeats itself. If the student Asserts something, then AutoTutor-1 gives evaluative feedback and advances the conversation with another dialog move.

In most turns, AutoTutor first *microadapts* to the student with short feedback and then *advances* the conversation further by selecting one of the substantive dialog moves (prompt, hint, assertion, correction, summary). But there are two other linguistic/discourse ingredients that are needed to make a smooth exchange. First, there a variety of discourse markers and frozen expressions that cue the student as to the discourse function of an ensuing dialog move. The discourse functions of hints, prompts, assertions, and corrections are quite distinct, but the student will be confused if the functions are left vague. Second, the final dialog act in a turn must make it perfectly clear to the student what to do next. It is best to end AutoTutor's turn with a prompt or question so that the student knows what to do next. Otherwise, the student will sit with a blank stare, not knowing who has the floor.

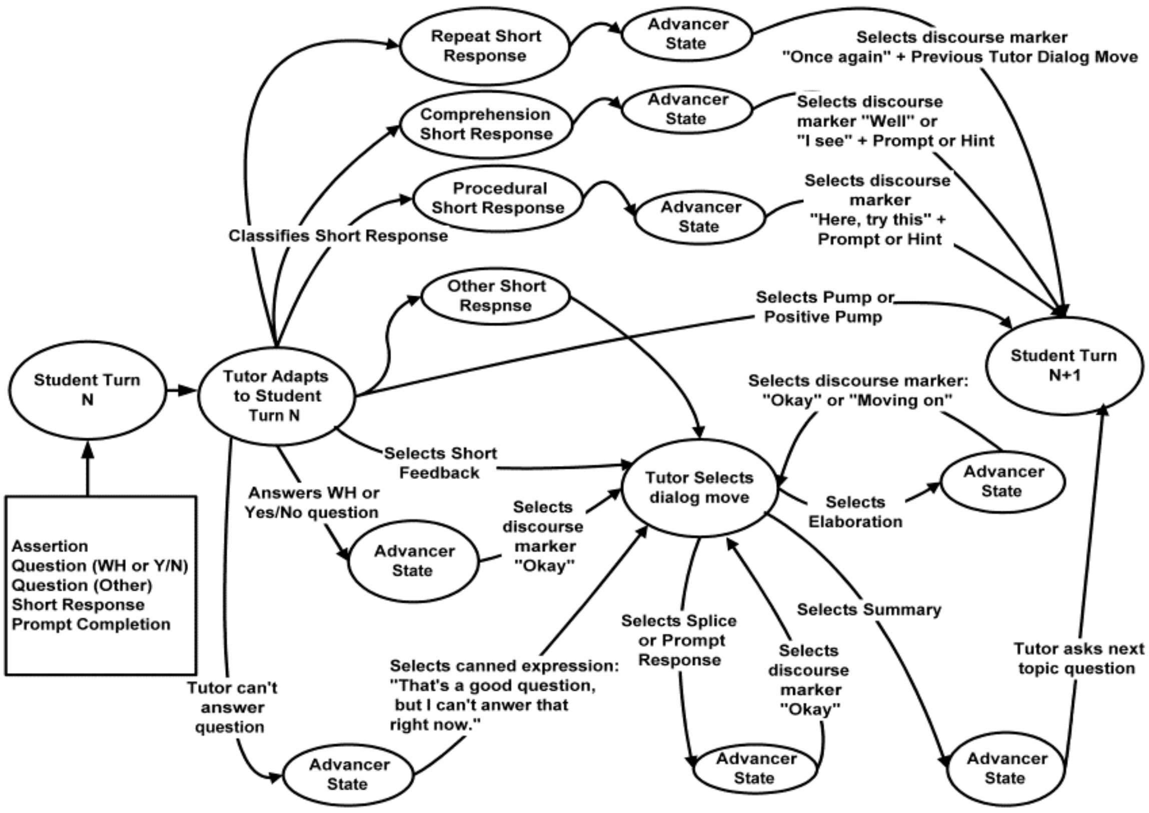

Repeat Short Response
Advancer State
Selects discourse marker "Once again" + Previous Tutor Dialog Move
Comprehension Short Response
Advancer State
Selects discourse marker "Well" or "I see" + Prompt or Hint
Procedural Short Response
Advancer State
Selects discourse marker "Here, try this" + Prompt or Hint
Classifies Short Response
Other Short Respnse
Selects Pump or Positive Pump
Student Turn N+1
Student Turn N
Tutor Adapts to Student Turn N
Selects discourse marker: "Okay" or "Moving on"
Advancer State
Selects Short Feedback
Tutor Selects dialog move
Selects Elaboration
Answers WH or Yes/No question
Advancer State
Selects discourse marker "Okay"
Assertion
Question (WH or Y/N)
Question (Other)
Short Response
Prompt Completion
Selects Summary
Selects Splice or Prompt Response
Selects discourse marker "Okay"
Tutor asks next topic question
Selects canned expression: "That's a good question, but I can't anwer that right now."
Tutor can't answer question
Advancer State
Advancer State
Advancer State

Figure 2. Dialogue Advancer Network (DAN)

Some example dialogs will illustrate some of the disasters that can occur when the DAN is absent or flawed. The current DAN has solved nearly all of AutoTutor's turn-taking problems in the useability tests. But our earlier versions were not perfect. The dialog excerpts below were collected in an alpha test in the early stages of AutoTutor. There are three parties in these interactions: AutoTutor, the learner, and a dedicated lab assistant who was immediately available to answer questions. It should be noted that there is no lab assistant dedicated to a single student in beta tests and field tests of AutoTutor. In Excerpts 1 and 2, the topic is parts and uses of a computer. In Excerpt 3, AutoTutor is discussing the features of RAM. AutoTutor's dialog moves are labeled in parentheses; the information in brackets refers to the students' utterances that were directed to a human lab assistant during alpha testing.

Excerpt 1

1.1 ***AutoTutor*** (Prompt): See if you can get this. Computers can be used to analyze data by finding averages, standard deviations, or performing many other kinds of . . .?
1.2 ***Student to lab assistant***: [Oh, just answer the question? Um. So, am I still supposed to answer that question (pointing to the main topic question that remains on the monitor)?]
1.3 ***Student types***: i don't know
1.4 ***AutoTutor*** (Prompt Response): kinds of statistics.
1.5 ***Student to lab assistant***: [Okay, this is really confusing. I don't understand what the question is supposed to be.]

1.6 ***Student types***: i don't know

Excerpt 2

2.1 ***AutoTutor*** (Assertion): Also, a computer has a monitor.
2.2 ***Student to lab assistant***: [I just answer yes or no?]
2.3 ***Student types***: yes

Excerpt 3

3.1 ***AutoTutor*** (Assertion): And, the CPU uses RAM as a short-term memory.
3.2 ***Student to lab assistant***: [Should I, like, I can just say how I think he's unintelligible?]
3.3 ***Student types***: i am not sure
3.4 ***AutoTutor*** (Neutral Feedback): Uh-huh
3.5 ***AutoTutor*** (Assertion): Also, RAM's contents disappear when the computer is turned off.
3.6 ***Student types***: what?
3.7 ***AutoTutor*** (Neutral Feedback): Uh-huh
3.8 ***AutoTutor*** (Hint): So, how does the CPU get information from RAM?
3.9 ***Student types***: I am not sure
3.10 ***AutoTutor*** (Neutral Feedback): Uh-huh
3.11 ***AutoTutor*** (Summary): So, RAM is primary memory storage and is changeable. It is used by the CPU as short-term memory. The CPU both writes to and reads from RAM. The contents of RAM disappear when the computer is turned off.

These three excerpts highlight several of AutoTutor's original conversational deficits. One glaring deficit involves the students' uncertainty about when it is their turn to respond to AutoTutor. Turn-taking is an integral feature of the conversational process. To facilitate the turn-taking process in human-to-human conversations, speakers signal to listeners that they are relinquishing the floor, i.e., it is the listener's turn to say something (Clark,, 1996; Hobbs, 1979; Nofsinger, 1991; Sacks et al., 1978). However, human-to-computer conversations lack many of the subtle paralinguistics signals inherent in human conversations. When conversational agents like AutoTutor lack turn-taking signals, computer users often do not know when or if they are supposed to respond. In conversations with AutoTutor, students were frequently confused after AutoTutor's Assertions (equivalent to Elaborations in Figure 2), after Prompt Responses, and after those Hints presented in declarative mood. They were confused because they did know what the function of the speech act is. When Hints were in interrogative mood (question form), they were not problematic for students. Most of these problems have been corrected by preceding the substantive dialog moves with appropriate discourse markers and frozen expressions that signal their functions. Sometimes the talking head needed to give gestures to clarify the function of the discourse move.

It should be noted that the DAN is a departure from more intelligent dialog management systems that involve dynamic planning (Freeman, 1999; Moore, 1995; Rich & Sidner, 1998). Such systems perform dynamic planning of dialogue moves on the basis of knowledge states, goals, and beliefs that the tutor infers about the student and that the tutor believes is in the common ground. Unfortunately, the language of learners is extremely vague and underspecified so the application of these models may have limited value. The primary challenge in discourse management in tutorial dialog is managing vague assertions of students rather than accurately inferring specific knowledge states of learners. In fact, Graesser et al.'s (1995) in depth analysis of human tutorial dialog revealed that tutors have only a crude, approximate sense of what students know.

In a recent analysis, we examined how AutoTutor utilized the DAN while interacting with students. More specifically, we wanted to document AutoTutor's DAN pathway choice for each student turn. Sixty-four students enrolled in a computer literacy course agreed to interact with AutoTutor in exchange for course credit. AutoTutor covered 24 computer literacy topics during each of the tutoring sessions and written transcripts were generated for all of the sessions. Three of the 24 computer literacy topics were randomly selected from each of the 64 transcripts. Thus, 192 mini-conversations were included in the DAN analysis.

The frequency distribution for the most well-traveled pathways is provided in Table 1. All pathways with frequencies lower than 10 are not included in the table. We were somewhat encouraged in that AutoTutor utilized 30 of the 78 legal DAN pathways and that there were no instances of illegal pathways after student turns. It is clearly the case, however, that the current version of AutoTutor is not maximizing the DAN to its full potential and that adjustments need to be made to break some of AutoTutor's poor conversational habits. Two common problems will illustrate this.

Table 1. Frequency Distribution of DAN Pathways Chosen by AutoTutor-1

DAN Pathway	*frequency*
Prompt Response → Advancer → Prompt	215
Positive Feedback → Prompt Response → Advancer → Prompt	179
Pump	169
Comprehension Short Response Advancer → Prompt	133
Repeat Short Response Advancer → Advancer	81
Neutral Feedback → Prompt	79
Prompt Response → Advancer → Summary	56
Positive Feedback → Prompt Response → Advancer → Summary	46
Prompt Response → Advancer → Elaboration → Advancer → Summary	37
Neutral Feedback → Hint	32
Positive Feedback → Prompt Response → Advancer → Elaboration → Advancer → Summary	26
Positive Feedback → Prompt Response → Advancer → Elaboration → Advancer → Prompt	10
Prompt Response → Advancer → Elaboration → Advancer → Prompt	10
*Total pathways	1134

*All pathways with frequencies below 10 are not included in the table.

The two most frequently traveled DAN pathways were the *Prompt Response → Advancer → Prompt* pathway (215 occurrences) and the *Positive Feedback → Prompt Response → Advancer → Prompt* pathway (179 occurrences). These two pathways alone comprised roughly 35% of all of the chosen pathways. In addition, 245 of the remaining pathways also ended in a Prompt. Hence, approximately 56% of the pathways ended with AutoTutor Prompting the student. This is problematic for two reasons. First, prompting a student at these rates is undesirable pedagogically. Human tutors usually reserve Prompts for medium to low ability students who are reluctant to provide any information. By relying on Prompts so often, AutoTutor is not giving students the opportunity to elaborate their knowledge about the topics.

Second, frequent prompting thwarts the conversational nature of AutoTutor that we are trying to promote. Prompts only require one or two word responses from students rather lengthier contributions that frequently occur in conversations. We have attempted to fix the Prompt problem by altering some of the pathways in the DAN and by modifying the conditions that trigger Prompts.

AutoTutor also has a problem with some of the short-immediate feedback pathways. Specifically, AutoTutor never selects the DAN pathways that include three of the short feedback categories (i.e., positive-neutral, negative-neutral, and negative). It would be wonderful if AutoTutor chose to ignore these pathways because all of the student Assertions were high in quality. Unfortunately, this was not the case. Many of the student Assertions contained misconceptions and a number of others were only partially correct. We have addressed the feedback problem by revising some of the dialog management components discussed below.

Fuzzy production rules

The selection of a dialog move category is sensitive to various parameters that are induced from the dialog history. The selection of the next category is determined by a set of 15 fuzzy production rules (Kosko, 1992). Fuzzy production rules are tuned to (a) the quality of the student's assertions in the preceding turn, as computed by LSA, (b) global parameters that refer to the ability, verbosity, and initiative of the student, and (c) the extent to which the good answer aspects of the topic had been covered. For example, consider the following dialog move rules:

```
(1) IF [student Assertion match with a good answer aspect = HIGH or VERY HIGH]
        THEN [select POSITIVE FEEDBACK]

(2) IF [student ability = MEDIUM or HIGH
        & Assertion match with good answer aspect = LOW]
        THEN [select HINT]
```

In Rule 1, AutoTutor will provide Positive Feedback (e.g., *Right*) in response to a high quality student Assertion. In Rule 2, AutoTutor will generate a Hint to bring the relatively high ability student back on track (e.g., *What about the size of the programs you need to run?*). The dialog move generator currently controls the substantive dialog moves: Pump, Hint, Splice, Prompt, Prompt Response, Assertion, Summary, and five forms of immediate short-feedback (positive, positive-neutral, neutral, negative-neutral, and negative).

Selection of next good answer aspect (GAA)

As mentioned earlier, an answer to a deep-reasoning question consists of a set of good answer aspects (A_1, A_2, … A_n), all of which need to be covered during the tutorial dialog. The selection of the next GAA to cover was determined by the *zone of proximal development* in AutoTutor-1. AutoTutor-1 keeps track of the extent to which each aspect (A_i) has been covered as the dialog evolves for a topic. The coverage metric varies from 0 to 1 and gets updated as each Assertion is produced by the tutor or learner. LSA is used to compute the extent to which the various Assertions cover the particular aspects associated with a topic. If some threshold (t) is met or exceeded, then the aspect A_i is considered covered. AutoTutor-1 selects, as the next aspect to cover, the aspect that has the highest subthreshold coverage score. For example, suppose that the threshold is .70 and the LSA values are .35, .66, .87, .02, and .71 for aspects 1 through 5, respectively. The next GAA to select would be A_2 because its .66 LSA coverage value is below .70 and higher than the other aspects at subthreshold (.35 and .02). A_3 and A_5 are above threshold and therefore regarded as covered. Therefore, AutoTutor-1 builds on the fringes of what is known in the discourse space between the student and AutoTutor. A topic is finished when all of the GAA's have LSA coverage values that meet or exceed the threshold t.

AutoTutor-2 was designed to improve the pedagogical quality of the tutor over and above AutoTutor-1. AutoTutor-2 incorporated tactics that attempt to get the *student* rather than the *tutor* to articulate the GAA that was selected. Whereas AutoTutor-1 scored aspect (A_i) as covered if it was expressed by either the tutor or the student, AutoTutor-2 counted only what the student says when evaluating coverage; so if aspect (A_i) was not expressed by the student, it is not scored as covered at all. This forced the student to articulate the explanations in their entirety, an extreme form of constructivism. In essence, AutoTutor-2 makes the student do all of the talking, whereas AutoTutor-1 regarded points as being covered if either party articulated the information in a shared discourse space.

AutoTutor-2 was also designed to improve conversational smoothness. AutoTutor-1 could hop around from GAA to GAA without any attempt to be coherent. Therefore, AutoTutor-2 had an algorithm that considered three criteria for generating the next GAA: the zone of proximal development (as in AutoTutor-1), discourse coherence, and centrality. The discourse *coherence* criterion attempts to have the next GAA be coherently related to the previous GAA. That is, the next aspect (A_i) is selected that is most similar to the previous aspect that was covered. *Centrality* is an index of how connected an aspect is to other GAA's. AutoTutor-2 selects A_i if it has a high family resemblance (match) to the remaining uncovered aspects; that is, there will be an attempt to select an aspect that drags in the content of the remaining aspects to be covered. Whereas AutoTutor-1 capitalized on the zone of proximal development exclusively, AutoTutor-2 also considered conversational coherence and pivotal ideas when selecting the next good answer aspect to cover.

Articulation of the next good answer aspect

As discussed above, there are pedagogical advantages to having the student articulate knowledge (as in AutoTutor-2) rather than having the knowledge be delivered by the tutor (which often occurs in AutoTutor-1). Therefore, a good answer aspect (GAA) was considered covered in AutoTutor-2 only if it is articulated by the student. There also should be a pedagogical value in having the knowledge be articulated precisely, formally, and with appropriate symbolic expressions rather than in the informal language that is typical of conversation (Biber, 1988; Clark, 1996). It could be argued that precision, formalization, and symbolization are critical features of the learning process that partly explain the success of the PACT algebra tutor (Heffernan & Koedinger, 1998; Koedinger et al., 1997) and the Andes physics tutor (Gertner & VanLehn, 2000).

There were a variety of ways to get the student to articulate the knowledge in AutoTutor-2. One method was to have a larger family of hints and prompts associated with any particular GAA. Each hint or prompt was designed to elicit a different noun-phrase, prepositional phrase or clause in the GAA. In essence, the hints and prompts are selected until the missing constituents have been supplied by the student. A second method was to have the tutor model the articulation of the good answer and to persistently stay on a GAA until it was articulated by the student. In fact, AutoTutor-2 implemented up to two cycles of "hint-prompt-assertion" when extracting the constituents of a particular GAA. That is, a hint was first generated in tutor turn N, then a prompt in turn N+2, then an assertion in N+4, and then additional cycles until all of the content was articulated by the student. This principle of *progressive specificity* in hinting mechanisms has been implemented in the ANDES physics tutor and in the PACT algebra tutor, but these systems have not yet finished implementing tutorial dialog in natural language. One method of getting the student to articulate the content more precisely is to raise the LSA threshold (t) for coverage of a GAA. As the threshold approaches 1.0, the student would be expected to articulate the information in a fashion that more closely matches the exact wording in the GAA.

Talking head with gestures

Researchers have recently developed computer-generated animated talking heads that have facial features synchronized with speech and in some cases appropriate gestures (Cassell & Thorisson, 1999, Cohen & Massaro, 1994; Johnson, Rickel, & Lester, 2000). Ideally, the computer controls the eyes, eyebrows, mouth, lips, teeth, tongue, cheekbones, and other parts of the face in a fashion that is meshed appropriately with the language and emotions of the speaker Picard, 1997). A talking head is an important feature of AutoTutor because it concretely grounds the conversation between the tutor and learner. A talking head also provides a separate channel of cues for providing mixed feedback to the learner. When a learner's contribution is incorrect or vague, for example, the speech is often positive and polite whereas the face has a puzzled expression; this conflicting message that satisfies both pedagogical and politeness constraints would be preferable to a threatening speech message that says "That's wrong" or "I'm having trouble understanding you." The nonverbal facial cues are known to be an important form of backchannel feedback during tutoring (Fox, 1993; Graesser et al., 1995), as well as other contexts of conversation (Clark, 1996). Similarly, pitch, pause, duration, amplitude, and intonation contours are among the intonation cues that signal backchannel feedback, affect, and emphasis (Brennan & Williams, 1995).

AutoTutor's dialog moves are delivered by a talking head that synchronizes synthesized speech, facial expressions, and sometimes gestures. Microsoft Agent is currently being used as the talking head with synthesized speech, with parameters of the facial expressions and intonation being generated by fuzzy production rules (McCauley, Gholson, Hu, Graesser, & TRG, 1998). Unfortunately, the grain size of the intonation and facial parameters of Microsoft Agent is too crude to handle the subtle facial expressions that we desire. Also, the sequential constraints of microcomputers make it difficult to handle the synchronization of components in parallel. For example, the current version of Microsoft Agent does not allow AutoTutor to point and to display facial expressions at the same time that it produces synthesized speech. Therefore, AutoTutor-2 has a version that uses Java 3D plus a neurofuzzy controller to allow lower-level programming for new behaviors on the fly, after interpolating from a sample of pre-scripted prototype behaviors during its development. The animated agents in the MIT Media Labs (Cassell & Thorisson, 1999) normally require several powerful computers to provide synchronization of speech, intonation, face, and gesture, whereas AutoTutor-2 is delivered on a Pentium.

EVALUATIONS OF AUTOTUTOR

AutoTutor has been tested on nearly 200 students in a computer literacy course at the University of Memphis. The tutoring was provided as extra credit in the course at a point in time after the students had allegedly read the relevant chapters and attended a lecture in the course. So AutoTutor gave students an opportunity to have additional study of the material. Our evaluations of gains in learning and memory were very promising. AutoTutor provided an effect size increment of .5 to .6 SD units when compared to control conditions (Graesser, Bautista, et al., 2001; Graesser, VanLehn, et al., in press; Link et al., 2000). This increment in learning was found for test questions that tap both deep and shallow learning. These results are on par, if not better, than the .4 SD effect size that occurs in normal human tutoring (Cohen et al., 1982).

In order to illustrate our methods of assessing learning gains, consider one of the experiments that we conducted on AutoTutor-1 (Graesser et al., 2001). AutoTutor-1 was tested on 36 students in a computer literacy class. Each student had the three macrotopics (hardware, operating systems, Internet) assigned to a different condition, using a suitable counterbalancing scheme: *AutoTutor* (student uses AutoTutor to study one of the macrotopics), *Reread* (student re-reads a chapter associated with a different macrotopic), and no-read *Control* (student does not re-study the remaining macrotopic). A repeated measures design was used so that we could evaluate Aptitude x Treatment interactions; that is, we could assess whether AutoTutor is relatively effective for some categories of learners but not others (such as high versus low performers overall). On the average, students took 38 minutes to use AutoTutor, which was somewhat less time than the 45 minutes assigned in the Reread condition. There were 3 outcome measures. There was a sample of testbank questions that were actually used in the computer literacy course; these were in an N-alternative multiple-choice format . We discovered that all of these questions were *shallow* according to Bloom's taxonomy of cognitive difficulty (Bloom, 1956). There was a sample of *deep* multiple choice questions, one question for each of the 36 topics, that tapped causal inferences and reasoning. And finally, there was a cloze test that had 4 critical words deleted from the ideal answers of each topic; the students filled in these blanks with answers. The proportion of correct responses served as the metric of performance. We also combined all three outcome measures into a composite score. There were significant differences in composite scores among the three conditions, with means of .43, .38, and .36 in the AutoTutor, Reread, and Control conditions, respectively, $\underline{F}(2, 70) = 6.10$, $\underline{p} < .05$. Planned comparisons showed the following pattern: AutoTutor > Reread = Control. The effect size of AutoTutor over Control was .5. A repeated measures ANOVA was performed that crossed the three conditions with the three types of subtests. There was a significant main effect of condition, $\underline{F}(2, 70) = 48.03$, $\underline{p} < .05$, MSe = .038, a significant main effect to test, $\underline{F}(2, 70) = 3.06$, $\underline{p} < .05$, MSe = .037, but no significant interaction. Aptitude x Treatment interactions were not found in this study but we remain in the hunt for such interactions. These results support the conclusion that AutoTutor had a significant impact on learning gains.

We have evaluated AutoTutor on the conversational smoothness and the pedagogical quality of its dialog moves in the turn-by-turn tutorial dialog (Person, Graesser, Kreuz et al., in press). When experts rate the quality of AutoTutor's dialog moves, the mean ratings are positive (i.e., smooth rather than awkward, good rather than bad pedagogical quality), but there clearly is room to improve in the naturalness and pedagogical effectiveness of its dialog. In a recent study performed in our lab, we performed a *bystander Turing test* on the naturalness of AutoTutor's dialog moves. We randomly selected 144 tutor moves in the tutorial dialogs between students and AutoTutor-1. We asked 6 human tutors (from the tutor pool on computer literacy at the University of Memphis) to fill in what they would say at these 144 points. So at each of these 144 tutor turns, we had what the human tutor generated and what AutoTutor generated. We subsequently tested a group of 36 computer literacy students as to whether they could discriminate between dialog moves that were generated by a human versus a computer; half in fact were by human and half were by computer. We found that these students were unable to discriminate whether particular dialog moves had been generated by a computer versus a human; the *d'* discrimination scores were actually a bit negative (-.08), but not significantly. This rather impressive outcome supports the claim that AutoTutor is a good simulation of human tutors.

AutoTutor has done a surprisingly good job evaluating the quality of the answers that students type in during the tutorial dialog. AutoTutor attempts to "comprehend" the student input by segmenting the contributions into speech acts and matching the student's contributions to good answer aspects and bad answers through LSA (Landauer & Dumais, 1997). Our research revealed that AutoTutor is almost as good as an expert in computer literacy in evaluating the quality of student answers to questions and the quality of contributions in the tutorial dialog (Graesser et al., 2000; Wiemer-Hastings et al., 1999). For example, 2 graduate student research assistants had a correlation of approximately .5 to .6 when grading the quality of student answers, whereas there is nearly a .5 correlation between AutoTutor's LSA component and a graduate student RA. Some critics may not be impressed with the .5 to .6 interjudge reliability scores, but it should be noted that the interjudge reliability correlations were approximately .6 to .7 when Foltz (1996) had expert composition teachers grade essays. The goal of this research is not to carefully train a group of experts to optimize their reliability scores (a goal of some research projects). Instead, our goal is to obtain a reasonable estimate of the reliability of these scores in a naturalistic context and to observe how well AutoTutor's LSA component compares. We found that our LSA evaluator of the quality of student contributions was in the arena of graduate student RA's, the individuals who normally grade these answers in a university course.

Colleagues frequently ask how students respond emotionally to the talking head of AutoTutor. Unfortunately, we have not performed a systematic evaluation of the students' emotions, but we do have a number of impressions after having run approximately 200 students. Most students are initially amused by the talking head, but the amusement wears off in a few minutes. Some of the students initially have trouble understanding the synthesized speech, but all of these students adjust within about 5 minutes and can comprehend AutoTutor. All of the students have found AutoTutor sufficiently engaging to complete the tutorial sessions. A minority of the students occasionally become irritated when AutoTutor's speech acts are inappropriate or when the student thinks AutoTutor is not listening at a deep enough level. Of course, we would expect a diverse array of emotional responses to any new communication technology. Additional research is needed to assess the emotional response of students after they work with AutoTutor for several hours, days, and weeks. Additional research is also needed to assess how much of the learning gains can be attributed to the talking head versus the other language and discourse modules of AutoTutor.

CLOSING COMMENTS

The success of AutoTutor rests on the fundamental premise that discourse patterns provide an important class of teaching tactics and strategies. A computer tutor can be viewed as a dialogue partner that assists the learner in exploring his or her own knowledge and that exposes the students to the constraints of the problem and the fragments of the good answer. In essence, AutoTutor is a discourse prosthesis that scaffolds the student to new levels of mastery through conversation. We believe that one important key to learning lies in getting the student to say the right thing and the right time.

ACKNOWLEDGEMENTS

Research on AutoTutor was supported on grants from the National Science Foundation (SBR 9720314) and the Office of Naval Research (N00014-00-1-0600).

REFERENCES

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

Biber, D. (1988). *Variations Across Speech and Writing*. Cambridge: Cambridge University Press.

Biggs, J. (1996). Enhancing teaching through constructive alignment. *Higher Education*, 32, 347-364.

Bloom, B.S. (1956). *Taxonomy of Educational Objectives: The Classification of Educational Goals. Handbook I: Cognitive Domain.* New York: McKay.

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.

Bransford, J. D., Goldman, S. R., and Vye, N. J. (1991). Making a difference in people's ability to think: Reflections on a decade of work and some hopes for the future. In R. J. Sternberg and L. Okagaki (Eds.), *Influences on Children* (pp. 147-180). Hillsdale, NJ: Erlbaum.

Brennan, S. E., and Williams, M. (1995). The feeling of another's knowing: Prosody and filled pauses as cues to listeners about the metacognitive states of speakers. *Journal of Memory and Language*, 34, 383-398.

Brown, A. L. (1988). Motivation to learn and understand: On taking charge of one's own learning. *Cognition and Instruction*, 5, 311-321.

Cassell, J., and Thorisson, K.R. (1999). The power of a nod and a glance: Envelope vs. emotional feedback in animated conversational agents. *Applied Artificial Intelligence*, 13, 519-538.

Chi, M. T. H., de Leeuw, N., Chiu, M., and LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science*, 18, 439-477.

Clark, H.H. (1996). *Using Language*. Cambridge: Cambridge University Press.

Cohen, M.M., and Massaro, D.W. (1994). Development and experimentation with synthetic visible speech. *Behavior Research Methods, Instruments, and Computers*, 26, 260-265.

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

Collins, A. (1985). Teaching reasoning skills. In S.F. Chipman, J.W. Segal, and R. Glaser (Eds), *Thinking and Learning Skills* (vol. 2, pp 579-586). Hillsdale, NJ: Erlbaum.

Collins, A., Brown, J. S., and Newman, S. E. (1989). Cognitive apprenticeship: Teaching the craft of reading, writing, and mathematics. In L. B. Resnick (Ed.), *Knowing, Learning, and Instruction: Essays in Honor of Robert Glaser* (pp. 453-494). Hillsdale, NJ: Erlbaum.

Corbett, A., Anderson, J., Graesser, A., Koedinger, K., and van Lehn, K. (1999). Third generation computer tutors: Learn from or ignore human tutors? *Proceedings of the 1999 Conference of Computer-Human Interaction* (pp. 85-86). New York: ACM Press.

Core, M.G., Moore, J.D., and Zinn, C. (2000). Supporting constructive learning with a feedback planner. In *Papers from the 2000 AAAI Fall Symposium* (pp, 1-9). Menlo Park, CA: AAAI Press.

Du Boulay, B. (2000). Can we learn from ITSs? *International Journal of Artificial Intelligence in Education*, 11, 1040-1049.

Foltz, P.W. (1996). Latent semantic analysis for text-based research. *Behavior Research Methods, Instruments, and Computers*, 28, 197-202.

Fox, B. (1993). *The Human Tutorial Dialog Project*. Hillsdale, NJ: Erlbaum.

Freedman, R. (1999). Atlas: A plan manager for mixed-initiative, multimodal dialogue. *Proceedings of the AAAI-99 Workshop on Mixed Initiative Intelligence*.

Gagné, R. M. (1977). *The Conditions of Learning* (3rd ed.). New York: Holdt, Rinehart, & Winston.

Gertner, A.S., and VanLehn, K. (2000). Andes: A coached problem solving environment for physics. In G. Gauthier, C. Frasson, and K. VanLehn (Eds.), *Intelligent Tutoring Systems: 5th International Conference, ITS 2000* (pp. 133-142). New York: Springer.

Graesser, A.C., Bautista, L., Link, K., Kreuz, R., and the Tutoring Research Group (2001). An evaluation of AutoTutor on learning gains. Paper presented at the American Educational Research Association, Seattle, Washington.

Graesser, A.C., Franklin, S., and Wiemer-Hastings, P. and Tutoring Research Group (1998). Simulating smooth tutorial dialog with pedagogical value. *Proceedings of the American Association for Artificial Intelligence* (pp. 163-167). Menlo Park, CA: AAAI Press.

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

Graesser, A.C., Person, N.K., and Magliano, J.P. (1995). Collaborative dialog patterns in naturalistic one-on-one tutoring. *Applied Cognitive Psychology*, 9, 359-387.

Graesser, A.C., VanLehn, K., Rose, C., Jordan, P., and Harter, D. (in press). Intelligent tutoring systems with conversational dialogue. *AI Magazine*.

Graesser, A.C., Wiemer-Hastings, K., Wiemer-Hastings, P., Kreuz, R., and Tutoring Research Group (1999). AutoTutor: A simulation of a human tutor. *Journal of Cognitive Systems Research*, 1, 35-51.

Graesser, A.C., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Person, N., and Tutoring Research Group (2000). Using latent semantic analysis to evaluate the contributions of students in AutoTutor. *Interactive Learning Environments*, 8, 129-148.

Hacker, D.J., Dunlosky, J., and Graesser, A.C. (1998)(Eds.). *Metacognition in Educational Theory and Practice*. Mahwah, NJ: Erlbaum.

Heffernan, N.T., and Koedinger, K.R. (1998). The composition effect in symbolizing: The role of symbol production vs. text comprehension. *Proceedings of the 20th Annual Conference of the Cognitive Science Society* (pp. 307-312). Hillsdale, NJ: Erlbaum.

Hobbs, J. R. (1979). Coherence and coreference. *Cognitive Science*, 3, 67-90.

Hume, G. D., Michael, J.A., Rovick, A., and Evens, M. W. (1996). Hinting as a tactic in one-on-one tutoring. *The Journal of the Learning Sciences*, 5, 23-47.

Johnson, W. L., & Rickel, J. W., and Lester, J.C. (2000). Animated pedagogical agents: Face-to-face interaction in interactive learning environments. *International Journal of Artificial Intelligence in Education*, 11, 47-78.

Kalyuga, S., Chandler, P., and Sweller, J. (1999). Managing split-attention and redundancy in multimedia intsruction. *Applied Cognitive Psychology*, 13, 351-371.

Kintsch, W. (1998). *Comprehension: A Paradigm for Cognition*. Cambridge, MA: Cambridge University Press.

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

Kosko, B. (1992). *Neural Networks and Fuzzy Systems*. New York: Prentice Hall.

Landauer, T.K., and Dumais, S.T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review* , 104, 211-240.

Landauer, T.K., Foltz, P.W., and Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, 25, 259-284.

Lehmann, F. (1992)(Eds.). *Semantic Networks in Artificial Intelligence*. New York: Pergamon.

Lenat, D.B. (1995). CYC: A large-scale investiment in knowledge infrastructure. *Communications of the ACM*, 38, 33-38.

Lepper, M. R., Woolverton, M., Mumme, D.L., and Gurtner, J.L. (1991). Motivational techniques of expert human tutors: Lessons for the design of computer-based tutors. In S.P. Lajoie and S.J. Derry (Eds.), *Computers as Cognitive Tools* (pp. 75-105). Hillsdale, NJ: Erblaum.

Lesgold, A., Lajoie, S., Bunzo, M., and Eggan, G. (1992). SHERLOCK: A coached practice environment for an electronics troubleshooting job. In J. H. Larkin and R. W. Chabay (Eds.), *Computer-assisted Instruction and Intelligent Tutoring Systems* (pp. 201-238). Hillsdale, NJ: Erlbaum.

Link, K., Pomeroy, V., DiPaolo, R., Rajan, S., Klettke, B., Bautista, L., Kreuz, R., Graesser, A.C., and Tutoring Research Group (2000). The effectiveness of tutorial dialog in an automated conversational tutor. *Proceedings of the 10th Annual Meetings of the Society for Text and Discourse* (pp. 154-155). Lyon, France.

Marineau, J., Wiemer-Hastings, P., Harter, D., Olde, B., Chipman, P., Karnavat, A., Pomeroy, S., Graesser, A.C., and Tutoring Research Group (2000). Classification of speech acts in tutorial dialog. *Proceedings of the workshop on tutorial dialogue at the Intelligent Tutoring Systems 2000 conference*.

Mayer, R.E. (1997). Multimedia learning: Are we asking the right questions? *Educational Psychologist*, 32, 1-19.

McCauley, L., Gholson, B., Hu, X., Graesser, A.C., and the Tutoring Research Group (1998). Delivering smooth tutorial dialogue using a talking head. *Proceedings of the Workshop on Embodied Conversation Characters* (pp. 31-38). Tahoe City, CA: AAAI and ACM.

Mehan, H. (1979). *Learning Lessons: Social Organization in the Classroom*. Cambridge, MA: Harvard University Press.

Merrill, D. C., Reiser, B. J., Ranney, M., and Trafton, J. G. (1992). Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *The Journal of the Learning Sciences*, 2, 277-305.

Moore, J.D. (1995). *Participating in Explanatory Dialogues*. Cambridge, MA: MIT Press.

Moreno, R., and Mayer, R. E. (1999). Cognitve principles of multimedia learning: The role of modality and contingency. *Journal of Educational Psychology*, 91, 358-368.

Moshman, D. (1982). Exogenous, endogenous, and dialectical constructivism. *Developmental Review*, 2, 371-384.

Nofsinger, R. E. (1991). *Everyday Conversation*. Newbury Park, CA: Sage.

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

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books.

Person, N.K, and Graesser, A.C. (1999). Evolution of discourse in cross-age tutoring. In A.M. O'Donnell and A. King (Eds.), *Cognitive Perspectives on Peer Learning* (pp. 69-86). Mahwah, NJ: Erlbaum.

Person, N. K., Graesser, A. C., and the Tutoring Research Group (2000). Designing AutoTutor to be an effective conversational partner. *Proceedings for the 4th International Conference of the Learning Sciences*. Ann Arbor, MI.

Person, N.K., Graesser, A.C., Kreuz, R.J., Pomeroy, V., and TRG (in press). Simulating human tutor dialog moves in AutoTutor. *International Journal of Artificial Intelligence in Education*.

Person, N. K., Graesser, A. C., Magliano, J. P., and Kreuz, R. J. (1994). Inferring what the student knows in one-to-one tutoring: The role of student questions and answers. *Learning and Individual Differences*, 6, 205-29.

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

Piaget, J. (1952). *The Origins of Intelligence*. New York: International University Press.

Picard, R.W. (1997). *Affective Computing*. Cambridge, MA: MIT Press.

Pressley, M., and Wharton-McDonald, R. (1997). Skilled comprehension and its development through instruction. *School Psychology Review*, 26, 448-466.

Putnam, R. T. (1987). Structuring and adjusting content for students: A study of live and simulated tutoring of addition. *American Educational Research Journal*, 24, 13-48.

Rich, C., and Sidner, C.L. (1998). COLLAGEN: A collaborative manager for software interface agents. *User Modeling and User-adapted Interaction, 8*, 315-350.

Sacks, H., Schegloff, E. A., and Jefferson, G. (1978). A simplest systematics for the organization of turn taking for conversation. In J Schenkein (Ed.), *Studies in the Organization of Conversational Interaction*. New York: Academic Press.

Sleeman, D., and Brown, J. (1982)(Eds). *Intelligent Tutoring Systems*. New York: Academic Press.

Soller, A., Linton, F., Goodman, B., and Lesgold, A. (1999). Toward intelligent analysis and support of collaborative learning interaction. In S.P. Lajoie and M. Vivet, *Artificial Intelligence in Education* (pp. 75-82). Amsterdam: IOS Press.

Sweller, J., and Chandler, P. (1994). Why some material is difficult to learn. *Cognition & Instruction*, 4, 295-312.

VanLehn, K. (1990). *Mind Bugs: The Origins of Procedural Misconceptions*. Cambridge, MA: MIT Press.

VanLehn, K., Jones, R.M., and Chi, M.T. (1992). A model of the self-explanation effect. *The Journal of the Learning Sciences*, 2, 1-59.

Vygotsky, L.S. (1978). *Mind in Society*. Cambridge, MA: Harvard University Press.

Whittaker, S. (in poress). Mediated communication. In A.C. Graesser, M.A. Gernsbacher, and S.A. Goldman (Eds.), *Handbook of Discourse Processes*. Hillsdale, NJ: Erlbaum.

Wiemer-Hastings, P., Graesser, A.C., Harter, D., and the Tutoring Research Group (1998). The foundations and architecture of AutoTutor. *Proceedings of the 4th International Conference on Intelligent Tutoring Systems* (pp. 334-343). Berlin, Germany: Springer-Verlag.

Wiemer-Hastings, P., Wiemer-Hastings, K., and Graesser, A. (1999). Improving an intelligent tutor's comprehension of students with Latent Semantic Analysis. In S.P. Lajoie and M. Vivet, *Artificial Intelligence in Education* (pp. 535-542). Amsterdam: IOS Press.

Appendix A
An Example Topic in the Curriculum Script for One Question and Answer

\topic_Operating_System
\problem_solution-8 \moderate

\info-8 !Large, !multi-user !computers often work on several jobs !simultaneously. This is known
as !concurrent processing. Computers with state-of-the-art !parallel !processing use multiple CPUs to
process !several jobs simultaneously. However, the typical computer today has only !one CPU.
So here's your !question.

\question-8 How does the operating system of a typical computer process !several jobs simultaneously, with only !one CPU?

\ideal-8 The operating system helps the computer to work on several jobs simultaneously by rapidly switching back and forth between jobs. By rapidly switch back and forth between jobs, the operating system takes advantage of idle time on one job by working on another job. Timesharing computers use concurrent processing whenever multiple users are connected to the system. A timesharing computer moves from terminal to terminal, checking for input and processing each user's data in turn. Concurrent processing is common in personal computer operating systems that allow multitasking. Multitasking allows the computer user to issue a command that initiates a process in one application while the user works with other applications.

\pgood-8-1 The operating system helps the computer to work on several jobs simultaneously by rapidly switching back and forth between jobs.
\passert-8-1 The operating system switches rapidly !back and !forth between !jobs.
\phint-8-1-1 How can the operating system take advantage of !idle !time on the job?
\phintc-8-1-1 The operating system switches between jobs.
\phint-8-1-2 How does the operating system manage different programs?
\phintc-8-1-2 It switches rapidly back and forth.
\phint-8-1-3 Why would the operating system switch rapidly between jobs?
\phintc-8-1-3 The computer can work on several jobs simultaneously.
\pprompt-8-1-1 The operating system switches rapidly between
\ppromptc-8-1-1 Between !jobs.
\ppromptk-8-1-1 jobs.
\pprompt-8-1-2 Instead of being !sequential , the computer works on several jobs
\ppromptc-8-1-2 Several jobs !simultaneously.
\ppromptk-8-1-2 simultaneously, concurrently, at the same time.
\pprompt-8-1-3 Several jobs are run at the same !time by having the operating system
\ppromptc-8-1-3 The operating system !switch between jobs.
\ppromptk-8-1-3 switch, alternate.

\pgood-8-2 When there is idle time on one process or job, the operating system takes advantage of this
idle time by working on another job.
\passert-8-2 The operating system takes advantage of idle time on !one job by working on !another job.
\phint-8-2-1 How does the operating system avoid idle time on a job?
\phintc-8-2-1 When there is idle time on one process or job, the operating system works on another job.
\phint-8-2-2 How can the !operating system take advantage of !idle !time on a job?
\phintc-8-2-2 The operating system works on another job.
\phint-8-2-3 What computer component allows the computer to work on !several jobs at the same !time?
\phintc-8-2-3 The operating system.
\pprompt-8-2-1 When there is idle time on !one job, the operating system can work on another
\ppromptc-8-2-1 On another !job.
\ppromptk-8-2-1 job.
\pprompt-8-2-2 The operating system can work on !another job when the !first job encounters

\ppromptc-8-2-2 Encounters !idle !time.
\ppromptk-8-2-2 idle time.
\pprompt-8-2-3 A job faces idle time when there is no progress on any of its
\ppromptc-8-2-3 Any of its !processes.
\ppromptk-8-2-3 processes.

THERE ARE 3 ADDITIONAL GOOD ANSWER ASPECTS

\bad-8-1 The operating system completes one job first and then works on another.
\bbad-8-1 The operating system does one job at a time.
\splice-8-1 The operating system can work on !several jobs at !once.

\bad-8-5 If I give my command and someone else gives their command, then my command will be
carried out first.
\bbad-8-5 The command entered first is done first.
\splice-8-5 The operating system responds to !both commands !concurrently.

THERE ARE 5 ADDITIONAL BAD ANSWERS

\summary-8 The operating system !rapidly switches !back and !forth between !jobs. When there is
idle time on !one job, the operating system switches to !another job. multi !tasking allows the computer to work concurrently on !one command while processing !other commands of a single user.
!Timesharing allows several !users to use an operating system !simultaneously.