AutoTutor: A Cognitive System that Simulates a Tutor that Facilitates Learning through Mixed-initiative Dialogue

Arthur C. Graesser, Andrew Olney, Brian C. Haynes, and Patrick Chipman

University of Memphis

Send correspondence to:

Art Graesser
Psychology Department
202 Psychology Building
University of Memphis
Memphis, TN, 38152-3230
901-678-2742
901-678-2579 (fax)
a-graesser@memphis.edu

AutoTutor is a complex cognitive system that simulates a human tutor, or an ideal tutor, by holding a conversation with the learner in natural language. AutoTutor qualifies as cognitive system in two fundamental ways (see Forsythe and Xavier, this volume). First, its core architecture incorporates a human-like model of knowledge, discourse, and cognitive processes. Second, AutoTutor recruits these psychological mechanisms to engage the learner cognitively and thereby promote learning. Therefore, AutoTutor both simulates cognitive mechanisms of human tutors and facilitates cognitive activities of human learners.

AutoTutor was developed by Graesser and colleagues in the interdisciplinary Institute for Intelligent Systems at the University of Memphis (Graesser, Lu, Jackson, Hite-Mitchell, Ventura, Olney, & Louwerse, 2004; Graesser, Person, Harter, & TRG, 2001; Graesser, K. Wiemer-Hastings, P. Wiemer-Hastings, Kreuz, & TRG; 1999; Graesser, VanLehn, Rose, Jordan, & Harter, 2001). AutoTutor presents a series of challenging questions or problems that require approximately a paragraph (3-7 sentences) of information to answer correctly. An example question in conceptual physics is “When a car without headrests on the seats is struck from behind, the passengers often suffer neck injuries. Why do passengers get neck injuries in this situation?” Although a perfect answer to this question is approximately 3-7 sentences in length, the initial answers by actual human learners are typically only 1 word to 2 sentences in length. This is where tutorial dialogue is particularly helpful, whether the dialogue is provided by AutoTutor or a human tutor. AutoTutor assists the learner in the evolution of an improved answer that draws out more of the learner’s knowledge that is relevant to the answer. The dialogue between AutoTutor and the learner typically lasts 50-200 *turns* (i.e., the learner expresses something, then the tutor, then the learner, and so on). There is a *mixed-initiative* dialogue to the extent that each dialogue partner can ask questions and start new topics of discussion. AutoTutor serves as a *discourse prosthesis* that draws out more of what the learner knows and that adaptively improves answers to questions through collaboration in natural language.

In this chapter, we hope to accomplish three goals. First, we describe what AutoTutor does and clarify the scope of its functionality. Second, we briefly review the evidence that AutoTutor both facilitates learning and simulates a reasonably good conversation partner. Third, we describe the architecture of AutoTutor's complex system.

What does AutoTutor do?

Perhaps the best way to convey what AutoTutor is all about is through an example conversational dialogue. The conversation below is a conversation with a student on the topic of conceptual physics. This depicts the true output of AutoTutor, as opposed to serving as a hypothetical illustration of what an ideal AutoTutor might do. The conversation begins with a main question selected by AutoTutor from the repertoire of questions in its curriculum.

AutoTutor-1: The sun exerts a gravitational force on the earth as the earth moves in its orbit around the sun. Does the earth pull equally on the sun? Explain why.

Student-2: I don't know.

AutoTutor-3: Could you at least try to give me an answer? That way, we'll have something as a starting place to work from. How does Newton's third law of motion apply to this situation?

Student-4: What?

AutoTutor-5: Once again, how does Newton's third law of motion apply to this situation?

Student-6: Does Newton's law apply to opposite forces?

AutoTutor-7: Yes, right! Hum. There's still something missing from your explanation. Newton's Third Law refers to the forces exerted by one body on another ______?

Student-8: body

AutoTutor-9: You're amazing! Bravo! Let's move on. What can you say about the magnitude of the force that the earth exerts on the sun?

Student-10: What is the difference between a force and a vector?

AutoTutor-11: Interesting question. Perhaps, the following answer helps. Vector is an arrow drawn to scale used to represent a vector quantity. Does this answer help you in understanding the material?

Student-12: no.

AutoTutor-13: I might not have understood what you meant. Ask another conceptual physics question if you need more information.

Student-14: What is the difference between force and a vector?

AutoTutor-15: Force is any influence that tends to accelerate an object, a push or pull, measured in Newtons. Force is a vector quantity and vector is an arrow whose length represents the magnitude of a quantity and whose direction represents the direction of the quantity. Okay, I bet you can get this! The sun experiences the same magnitude of gravitational pool from the earth, as the earth does from the _____?

Student-16: sun

AutoTutor-17: Yes, right! From the sun. What is going on in this situation between the earth and sun?

Student-18: The sun is pulling on the earth because it is bigger.

AutoTutor-19: Let's try this together. When two objects of different masses hit each other, which one exerts the greater force on the other, the heavy object, the light object, or are the forces equal?

Student-20: the heavy object.

AutoTutor-21: No. The forces are equal in magnitude. Um, well, for every force there is a reaction force, such that the two forces constitute an action reaction force pair. The two forces have opposite direction and equal magnitude. No matter how different the masses of the two objects are, they will exert reaction forces on each other in the same magnitude.

This example shows that AutoTutor can generate a reasonably smooth conversation. AutoTutor’s answers are adaptive to what the student previously said and appear to be interpreting the student’s speech acts. Obviously the dialogue is not perfect. One glitch of AutoTutor, for example, appears to be in turn 7. It is not clear whether AutoTutor is answering the student’s question that was asked in turn 6, or is incorrectly interpreting turn 6 as an assertion made by the student. However, for the most part, the conversation is reasonably smooth. It should be noted that the 21 turns in this example exchange does not completely answer the main question posed by AutoTutor in turn 1. We have conducted experiments on over 1000 students using AutoTutor, or roughly 2-3 thousand hours of logged time of conversational dialogue. Some students require several hundreds of turns to answer a single main question posted by AutoTutor.

AutoTutor needs to accomplish a number of tasks in order to orchestrate a smooth and adaptive conversation. AutoTutor needs to classify and interpret the language expressed by the student. Students type in their input at present, but we do have a prototype version that incorporates speech recognition. AutoTutor needs to formulate one or more dialogue moves within each conversational turn in a fashion that is responsive to the student. Table 1 presents an analysis of the example dialogue by specifying the categorized dialogue moves of AutoTutor, the classified speech acts of the student, and comments to help the reader interpret what is going on. The content expressed by either AutoTutor or the student in Table 1 are signified in italics. Discourse categories of AutoTutor's dialogue moves have been added in capitals, whereas other information is added in normal font.

Insert Table 1 about here

The example and analysis in Table 1 illustrate the mixed-initiative dialogue of AutoTutor. AutoTutor segments the student's turns into speech act units and then assigns these units into categories, such as Assertion, Short Answer, Metacognition, Metacommunication, Verification Question, and Comparison Question. There are approximately 20 categories of student speech acts; 16 of these are different categories of student questions. AutoTutor attempts to accommodate virtually any student question, assertion, comment, or extraneous speech act. This requires interpreting the student's speech acts by implementing modules developed in the field of computational linguistics (Allen, 1995; Covington, this volume; Jurafsky & Martin, 2000). AutoTutor needs to produce language in addition to comprehending language. Each turn of AutoTutor requires the generation of one or more dialogue moves that either adaptively respond to what the student just expressed or that advance the conversation in a constructive fashion that answers the main question. The dialogue moves within a turn are connected by dialogue markers, as illustrated in Table 1. Some dialogue moves are very responsive to the student's preceding turn, such as the short feedback (positive, neutral, versus negative), the answers to student questions, and corrections of student misconceptions. Other dialogue moves push the dialogue forward in an attempt to cover the expected answer to the questions. These forward-directed dialogue moves include Pumps (e.g., *Tell me more, What else?*), Hints, Prompts for specific words or phrases, and Assertions. The responsive and forward-directed dialogue moves together provide a mixed-initiative dialogue in which both parties of the conversation exert an influence over the conversation. These are not scripted conversations, but rather are dynamically emerging exchanges.

AutoTutor delivers its dialogue moves with an animated conversational agent that has a text-to-speech engine, facial expressions, gestures, and pointing. Animated agents have been become increasingly popular in learning environments on the web, Internet, and desktop applications (Cassell & Thorisson, 1999; Massaro & Cohen, 1995; Johnson, Rickel, & Lester, 2000). Figure 1 shows the interface of one version of AutoTutor on the subject matter of Newtonian physics. The main question is presented in the top-right window. This major question (e.g., involving a boy dropping keys in a falling elevator) remains at the top of the web page until it is finished being answered during a multi-turn dialogue. The students use the bottom-right window to type in their contributions for each turn. The dialogue history between AutoTutor and student is shown in the bottom-left window. The animated conversational agent resides in the upper-left area. The agent uses a text-to-speech engine from either AT&T, SpeechWorks, or a Microsoft Agent (dependent on licensing agreements) to speak the content of AutoTutor's turns. Figure 2 shows a somewhat different interface that is used when tutoring computer literacy. This interface has a display area for diagrams, but no dialogue history window.

Insert Figures 1, 2, and 3 about here

AutoTutor with interactive 3D simulation

One version of AutoTutor has an embedded interactive 3D simulation. 3D simulation provides an additional channel of communication to discuss conceptual physics with the learner. Each simulation is crafted to cover particular physics principles in an ideal answer or to help correct particular misconceptions about physics. For each of the physics problems, we developed an interactive simulation world in *3-d Studio Max*. This included the people, objects, and spatial setting associated with the problem. The student can manipulate parameters of the situation (e.g., mass of objects, speed of objects, distance between objects) and then ask the system to simulate what will happen. They can compare their expected simulated outcome with the actual outcome after the simulation is completed. Moreover, they describe what they see. Their actions and descriptions are evaluated with respect to covering the expected principles in an ideal answer. In order to manage the interactive simulation, AutoTutor gives hints and suggestions, once again scaffolding the learning process with dialogue. Thus, AutoTutor combines interactive simulation with mixed-initiative dialog.

Figure 3 shows an example interface for the 3D version of AutoTutor. The question is presented at the top of the screen, in this case “When a car without headsets on the seats is struck from behind, the passengers often suffer neck injuries. Why do passengers get neck injuries in this situation?” Beneath the question are two windows that show the car and truck (middle window) and the driver in the car (right window). These components move whenever a simulation is run. Beneath the question on the left is the animated agent that guides the interaction with hints, suggestions, assertions, and other dialogue moves. These suggestions include having the student manipulate parameters, such as truck speed, mass of the car, and mass of the truck. The students also have a number of the binary options: Having the head rests in the car on, showing the skin on the driver, slowing down the simulation, and vector arrows that depict forces. The student manipulates these parameters and options, as shown in the bottom left, before a simulation is run. The activity of manipulating these inputs and viewing the simulation is believed to provide a referentially grounded and embodied representation of the problem, as well as a deeper understanding of physics (Bransford, Brown, & Cocking, 2000). However, empirical support for the pedagogical value of interactive simulation has not yet been substantiated in the literature on cognition and instruction. The students can run as many simulations as they wish until they feel they understand the relationship between parameters and outcomes of simulations. However, interacting with and viewing the simulations is not all there is. The participants are also prompted to describe what they see and answer the main question. Therefore, deep learning of physics is believed to emerge from the combination of interactivity, perceptual simulation, feedback on the simulation, and explaining what happens.

Some previous systems have conversational agents that combined dialogue with interactive simulation. Some examples of these systems are Steve (Rickel & Johnson, 1999) and Mission Rehearsal (Gratch, Rickel, Andre, Cassell, Petajan, & Badler, 2002). However, AutoTutor is the only system that is available on the Internet, that has systematically been tested on students, and that has flexible tutorial dialogue that scaffolds interactive simulation.

Pedagogical foundations of AutoTutor

The design of AutoTutor was inspired by three bodies of theoretical, empirical, and applied research. First, there are explanation-based constructivist theories of learning (Aleven & Koedinger, 2002; Chi, deLeeuw, Chiu, LaVancher, 1994; VanLehn, Jones, & Chi, 1992). According to the explanation-based constructivist theories of learning, learning is more effective and deeper when the learner must actively generate explanations, justifications, and functional procedures than when merely given information to read (Bransford et al., 2000). Second, there are intelligent tutoring systems that adaptively respond to student knowledge (Anderson, Corbett, Koedinger, & Pelletier, 1995; VanLehn, Lynch, et al., 2002). These tutors give immediate feedback to learner's actions and guide the learner on what to do next in a fashion that is sensitive to what the system believes the learner knows. Third, empirical research has documented the collaborative constructive activities that routinely occur during human tutoring (Chi, Siler, Jeong, Yamauchi, & Hausmann, 2001; Fox, 1993; Graesser, Person, & Magliano, 1995; Moore, 1995). The patterns of discourse uncovered in naturalistic tutoring are imported into the dialogue management facilities of AutoTutor.

One of AutoTutor's prominent dialogue patterns is called *expectation and misconception tailored dialogue* (EMT dialogue), which is known to be common in human tutoring. Both AutoTutor and human tutors typically have a list of anticipated good answers (called *expectations*) and a list of *misconceptions* associated with each main question or problem. One goal of the tutor is to coach the student in covering the list of expectations. Another goal is to correct misconceptions that are manifested in the student's talk and actions. Yet another goal is to give feedback and adaptively respond to the student. The expectations and misconceptions associated with a main question are stored in AutoTutor's *curriculum script.* AutoTutor provides *feedback* to the learner (positive, neutral, and negative feedback), *pumps* the learner for more information ("What else"), *prompts* the learner to fill in missing words, gives *hints*, fills in missing information with *assertions*, identifies and *corrects* bad answers, *answers* learners' questions, and *summarizes* answers. As the learner expresses information over many turns, the information in the 3-7 sentences of an expected answer is eventually covered and the question is answered. During the process of supplying the ideal answer, the learner periodically articulates misconceptions and false assertions. If these misconceptions have been anticipated in advance and incorporated into the program, AutoTutor provides the learner with information to correct the misconceptions. Therefore, as the learner expresses information over the turns, this information is compared to expectations and misconceptions, and AutoTutor formulates its dialogue moves in a fashion that is sensitive to the learner input.

AutoTutor does its best to handle questions posed by the learner. However, somewhat surprisingly, available research has revealed that students rarely ask questions in classrooms, human tutoring sessions, and AutoTutor sessions (Graesser & Person, 1994; Graesser & Olde, 2003). The rate of learner questions is 1 question per 6-7 hours in a classroom environment and 1 per 2 minutes in tutoring. This is disappointing news from the standpoint of a pedagogical theory that emphasizes curiosity and active inquiry. However, there is a silver lining. This characteristic of human-tutor interaction makes it easier to build a dialogue-based intelligent tutoring system such as AutoTutor. It is not computationally feasible to interpret any arbitrary input of the student from scratch and to construct a mental space that adequately captures what the learner has in mind. Instead, the best that AutoTutor can do is to perform conceptual pattern matching operations that compare student input with expectations. Fortunately, therefore, the prevailing tutorial activities between humans is compatible with what currently can be handled computationally within AutoTutor.

AutoTutor uses Latent Semantic Analysis (LSA) as its primary conceptual pattern matching algorithm when evaluating whether student input matches the expectations and misconceptions. LSA is a high-dimensional statistical technique that measures the conceptual similarity of any two pieces of text (Foltz, Gilliam, & Kendall, 2000; Kintsch, 1998; Landauer & Dumais, 1997; Laham, this volume; Landauer, Foltz, & Laham, 1998). The size may vary from being a single word to a sentence, paragraph, or lengthier document. A cosine is calculated between the LSA vector associated with expectation E (or misconception M) and the vector associated with learner input I. Expectation E (or misconception M) is scored as covered if the match between E or M and the learner's text input I meets some threshold, which has varied between .40 and .85 in previous instantiations of AutoTutor (Graesser, P. Wiemer-Hastings, K. Wiemer-Hastings, Harter, Person, & TRG, 2000; Olde, Franceschetti, Karnavat, Graesser, & TRG, 2002). As the threshold parameter increases, the learner needs to be more precise in articulating information and thereby cover the expectations.

Suppose that there are five key expectations in an ideal answer that the tutor wants to cover in an answer to a question. AutoTutor will direct the dialogue in a fashion that finesses the students to articulate each of these expectations. AutoTutor directs the dialogue through prompts, hints, and embedded 3D simulations that are targeted for particular expectations. AutoTutor stays on topic by completing the sub-dialogue that covers expectation E before starting a sub-dialogue on another expectation. For example, suppose an answer requires the expectation: *the force of impact will cause the car to experience a large forward acceleration.* The following family of prompts is available to encourage the student to articulate particular content words in the expectation:

1. The impact will cause the car to experience a forward _____?
2. The impact will cause the car to experience a large acceleration in what direction? _____.
3. The impact will cause the car to experience a forward acceleration with a magnitude that is very _____?
4. The car will experience a large forward acceleration after the force of ______?

5. The car will experience a large forward acceleration from the impact's ______?
6. What experiences a large forward acceleration?

The particular prompts that are selected are those that fill in missing information if answered successfully. That is, the dialogue management component adaptively selects hints, prompts, and targeted 3D simulations in an attempt to achieve pattern completion. The expectation is covered when enough of the ideas underlying the content words in the expectation are articulated by the student so that the LSA threshold is met or exceeded.

Once again, we believe that these dialogue mechanisms of AutoTutor are both computationally manageable and are very similar to what human tutors do. Human tutors cannot deeply comprehend all of the contributions of students, most of which are imprecise, vague, fragmentary, incomplete, and ungrammatical (Graesser et al., 1995). The best that most human tutors can do is to compare student input to anticipated good answers and misconceptions. The repertoire of anticipated content grows incrementally with tutoring experience. Comparisons between student input and anticipated content is approximate and scruffy rather than precise and tidy. LSA provides a suitable algorithm for such comparison operations. Moreover, the Expectation and Misconception Tailored (EMT) dialog moves of AutoTutor and most human tutors are not particularly sophisticated from the standpoint of ideal tutoring strategies that have been proposed in the fields of education and artificial intelligence (Graesser et al., 1995). Graesser and colleagues videotaped over 100 hours of naturalistic tutoring, transcribed the data, classified the speech act utterances into discourse categories, and analyzed the rate of particular discourse patterns. These analyses revealed that human tutors rarely implement intelligent pedagogical techniques such as *bona fide* Socratic tutoring strategies, modeling-scaffolding-fading, reciprocal teaching, frontier learning, building on prerequisites, or diagnosis/remediation of deep misconceptions (Collins, Brown, & Newman, 1989; Palincsar & Brown, 1984; Sleeman & Brown, 1982). These sophisticated techniques are summarized in Table 2. Instead of implementing these and many other sophisticated tutoring strategies, tutors tend to coach students in constructing explanations according to the EMT dialog patterns (strategy 1 in Table 2). The EMT dialog strategy is substantially easier to implement computationally than are the sophisticated tutoring strategies. On this dimension, the computational and psychological solutions are perfectly compatible.

Insert Table 2 about here

Adapting to learner emotions

We are in the process of developing a version of AutoTutor that perceives and responds to learner emotions in addition to the learner's knowledge states. AutoTutor is augmented with sensing devices and signal processing algorithms that classify affective states of learners. Emotions are classified on the basis of dialog patterns during tutoring, the content covered, facial expressions, body posture, mouse haptic pressure, and keyboard pressure. This recent project has two specific objectives. First, AutoTutor will analyze patterns of facial, body, and dialog activity that arise while interacting with AutoTutor and will classify this input into basic affect states (such as confusion, frustration, boredom, interest, excitement, and insight). Second, we will investigate whether learning gains and learner's impressions of AutoTutor are influenced by dialog moves of AutoTutor that are sensitive to the learner's emotions. For example, if the student is extremely frustrated, then AutoTutor presumably should give a good hint or prompt that directs the student in a more positive learning trajectory. If the student is bored, AutoTutor should give more engaging, challenging, and motivating problems. If the student is very absorbed and happy, then AutoTutor should be minimally invasive and stay out of the student's way.

There is already some evidence that emotions might be intimately interwoven with complex learning. We recently conducted an experiment in which we observed six different affect states (frustration, boredom, flow, confusion, eureka, and neutral) that potentially occur during the process of learning introductory computer literacy with AutoTutor (Craig, Graesser, Sullins, & Gholson, in press). The participants were 34 low-domain knowledge college students. Expert judges recorded emotions that learners apparently were experiencing at random points during the interaction with AutoTutor. Observational analyses revealed significant relationships between learning gains (posttest-pretest scores on multiple choice tests) and the affective states of boredom (r = -.39), flow (r = .29), and confusion (r = .33). Correlations with eureka (r = .03), and frustration (r = -.06) were near zero.

These results fit some available theoretical frameworks that interrelate emotions and cognition. The positive correlation between confusion and learning is somewhat provocative, but is actually consistent with a model that assumes that *cognitive disequilibrium* is one precursor to deep learning (Graesser & Olde, 2003; Otero & Graesser, 2001). Cognitive disequilibrium occurs when the learner experiences contradictions, discrepancies, novel input, obstacles to goals, decision deadlocks, and major knowledge gaps. Both cognitive activities and emotions are experienced until equilibrium is restored. The findings that learning correlates negatively with boredom and positively with flow are consistent with predictions from Csikszentmihalyi's (1990) analysis of *flow* experiences. Conscious flow occurs when the student is so absorbed in the material that time disappears, fatigue disappears, and extraneous interruptions get unnoticed. Experiences of eureka were much too rare in the experiment; there was only one recorded eureka experience in 17 total hours tutoring among the 34 students.

At this point in the project, we have assembled and installed most of the emotion sensing technologies with AutoTutor. We have analyzed the components, features, and representations of each of the sensing technologies (i.e., dialogue patterns during tutoring, content covered, facial expressions, body posture, mouse haptic pressure, and keyboard pressure). Software is currently being developed to interpret the input. These channels include (1) the AutoTutor log file with speech acts of student and tutor turns, as well as knowledge states achieved from the tutorial dialog, (2) the body posture pressure measurement system purchased from Tekscan, (3) the upper facial sensor device developed by Roz Picard's Affective Computing Lab at MIT (Picard, 1997; Kapoor & Picard, 2002), (4) a haptic pressure sensor for the mouse (supplied by MIT), and (5) a keyboard pressure sensor purchased from Tekscan. Affect states will be interpreted and/or classified on the basis of these five input channels of information. Computational models are being explored to perform these emotion analyses. These models have quantitative foundations in Bayesian, hidden Markov, neural network, and/or dynamical systems, but are substantially more complex than the standard architectures.

Empirical Evaluations of AutoTutor

AutoTutor should be declared a success to the extent that it meets various performance criteria. Four criteria have been considered in our previous evaluations of AutoTutor. One type is technical, and will not be addressed in this chapter. In essence, do particular computational modules of AutoTutor produce output that is valid and meets the intended technical specifications? We are satisfied, for example, that our LSA component performs conceptual pattern matching operations almost as well has human judges (Graesser, Hu, & McNamara, 2005; Graesser et al., 2000; Olde et al., 2002) and that our speech act and question classifier has a high degree of accuracy (Olney et al., 2003). A second type of evaluation assesses the quality of the dialogue moves produced by AutoTutor. That is, to what extent are AutoTutor's dialogue moves coherent, relevant, and smooth? A third criterion is whether AutoTutor produces learning gains. A fourth criterion is whether learners like interacting with AutoTutor. This section briefly presents what we know so far about the second and third types of evaluation.

Expert judges have evaluated AutoTutor with respect to conversational smoothness and the pedagogical quality of its dialogue moves (Person, Graesser, Kreuz, Pomeroy, & TRG, 2001). The experts' mean ratings lean to the positive end of the rating scales on conversational smoothness and pedagogical quality, but there is room to improve in the naturalness and pedagogical effectiveness of its dialogue. One intriguing evaluation has been a *bystander Turing test* on the naturalness of AutoTutor's dialogue moves (Person, Graesser, & TRG, 2002). In these studies, there was a random selection of tutor moves in the tutorial dialogs between students and AutoTutor. Six human tutors (from the tutor pool on computer literacy at the University of Memphis) were asked to fill in what they would say at these random points. At each of these random tutor turns, the corpus contained what the human tutors generated and what AutoTutor generated. A group of computer literacy students was asked to discriminate between dialogue moves generated by a human versus a computer; half in fact were by human and half were by computer. The results surprisingly revealed that the bystander students were unable to discriminate whether particular dialogue moves had been generated by a computer versus a human. The *d'* discrimination scores were near zero.

The above results of the bystander Turing test support the claim that AutoTutor is a good simulation of human tutors. AutoTutor manages to have productive and reasonably smooth conversations even though it does not completely understand what the student expresses. There is an alternative interpretation, however, which is just as interesting. Perhaps tutorial dialogue is not highly constrained, so the tutor has a high degree of latitude on what can be said without disrupting the conversation. In essence, there might be a large landscape of options on what the tutor can say at most points in the dialogue. The conversations are flexible and resilient, not fragile.

Evaluations of tutoring systems on learning gains is perhaps the most important performance criterion. It is well established that one-to-one human tutoring is a powerful method of promoting learning (Cohen, Kulik, & Kulik, 1982; Corbett, 2001), even though the vast majority of the human tutors have moderate domain knowledge and little or no training in pedagogy or tutoring. These unaccomplished human tutors enhanced learning with an effect size of .4 standard deviation units (called sigma's), which translates to approximately an improvement of half a letter grade. According to Bloom (1984), accomplished human tutors can produce effect sizes as high as 2-sigma in basic mathematics. However, the magnitude of this effect is suspect because only two studies have investigated the impact of accomplished tutors on learning gains. In the arena of computer tutors, intelligent tutoring systems with sophisticated pedagogical tactics, but no natural language dialog, produce effect sizes of approximately 1-sigma in the topics of algebra, geometry, and quantitative physics (Corbett, 2001; Van Lehn et al., 2002). Learning gains from tutors have been most pronounced on subject matters that are quantitative, with precise, clearcut answers.

AutoTutor has been evaluated on learning gains in several experiments on the topics of computer literacy (Graesser, Lu et al., 2004; Graesser, Moreno, Marineau, Adcock, Olney, Person, & TRG,. 2003; Person, Graesser, Bautista, Mathews, & Tutoring Research Group, 2001) and conceptual physics (Graesser, Jackson et al. 2003; VanLehn, Graesser, Jackson, Jordan, Olney, & Rose, 2004). The results of 9 experiments have been quite positive. Previous versions of AutoTutor have produced gains of .2 to 1.5 sigma (a mean of .8), depending on the learning performance measure, the comparison condition (either pretest scores or a control condition in which the learner reads the textbook for an equivalent amount of time as the tutoring session), the subject matter, and the version of AutoTutor. Approximately a dozen measures of learning have been collected in these assessments on the topics of computer literacy and physics, including: (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 subjects fill in missing words of texts that articulate explanatory reasoning on the subject matter, and (5) performance on problems that require problem solving. These results place previous versions of AutoTutor somewhere between an unaccomplished human tutor and an intelligent tutoring system. Moreover, one recent evaluation of physics tutoring remarkably reported that the learning gains produced by accomplished human tutors in computer mediated communication were equivalent to the gains produced by AutoTutor (Van Lehn et al., 2004). It is informative to note that the largest learning gains from AutoTutor have been on deep reasoning measures rather than measures of shallow knowledge. AutoTutor’s problems and dialogue facilities were designed to target deep reasoning so this result was quite expected.

One persistent question is what it is about AutoTutor that facilitates learning. For example, is it the dialogue content or the animated agent that explains the learning gains? What roles do motivation and emotions play, over and above the cognitive components? We suspect that the animated conversational agent will fascinate some students, and possibly be more motivating. Learning environments have only recently had animated conversational agents with facial features synchronized with speech and in some cases appropriate gestures (Cassell & Thorisson, 1999, Johnson, Rickel, & Lester, 2000). Many students will be fascinated with an agent that 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). The agents provide an anthropomorphic human-computer interface that simulates having a conversation with a human. This will be exciting to some, frightening to a few, annoying to others, and so on. There is some evidence that these agents tend to have a positive impact on learning or on the learner’s perceptions of the learning experience, compared with speech alone or text controls (Atkinson, 2002; Moreno, Mayer, Spires, & Lester, 2001). However, additional research is needed to determine the precise conditions, agent features, and levels of representation that are associated with learning gains. According to Graesser, Moreno et al. (2003), it is the dialogue content, not the speech or animated facial display, that influences learning, whereas the animated agent can have an influential role on motivation (positive, neutral, or negative). As expressed in Graesser, Moreno et al. (2002), “the medium is not the message – the message is the message.” Learning apparently is facilitated by a tutor that communicates the right content at the right time to the right student, whereas motivation is influenced by the aesthetics of the animated agent. One rather provocative result is that there is a near zero correlation between learning gains and how much the students like the conversational agents (Moreno, Klettke, Nibbaragandla, Graesser, & TRG, 2002). Therefore, it is important to distinguish liking from learning in this area of research. Although the jury is still be out on what it is exactly about AutoTutor that leads to learning gains, the fact is that students learn from the intelligent tutoring system and some enjoy having conversations with AutoTutor in natural language.

Architecture of AutoTutor

In the introductory chapter to this volume, Forsythe and Xavier argue that a cognitive system uses “plausible computational models of human cognitive processes as a basis for human machine interactions” (page 3). This is a distinctive conception of what an intelligent system interacting with a human should be, a stance for which we have considerable sympathy. Forsythe and Xavier propose two requirements for such a system: The system should posses an accurate model of both the user’s knowledge and the user’s cognitive processes. AutoTutor’s user modeling does in fact attempt to recover some of the knowledge and cognitive processes of the student learner. But it does so only statistically and coarsely, not symbolically and precisely. For example, LSA provides a statistical approximation of what the student knows about each expectation, as well as the student’s overall knowledge about the subject matter. AutoTutor’s conversation logs also keep track of how verbose the student is (measured as words per student turn) and how much the student takes the initiative (measured as the number of student questions). However, that is the limit on what AutoTutor knows about the student. This learner modeling of AutoTutor is intentionally minimal because of the vagueness and indeterminacy of student contributions and because we know there are limits to how much human tutors know about the students’ knowledge states (Graesser et al., 1995).

The computational model underlying AutoTutor is much more refined from the standpoint of its attempts to model what human tutors do. We indeed designed AutoTutor to incorporate the cognitive representations and processes of actual human tutors. For example, the

Expectation and Misconception Tailored (EMT) dialogues are distinctly modeled to match the dialogue patterns and pedagogical strategies of human tutors.

This section describes the overall computational architecture of AutoTutor. Given that many versions of AutoTutor have been developed and described (Graesser et al., 1999, 2001), we describe our most recent version that accommodates interactive 3D simulation. This version is called AutoTutor-3D, even though there is the option of removing the interactive 3D simulation module. We subsequently focus on one module that is particularly at the heart of AutoTutor, namely dialogue management.

Overall architecture of AutoTutor-3D

AutoTutor 3D is a client-server application on the Internet that uses a *thin client* protocol with the asynchronous transmission of small data packets (< 20K). In a dialogue turn, a *packet* is sent from the client (i.e., the computer or workstation that one learner interacts with) to the server (the central computer that handles many learners) and back again. These packets contain the global state of the tutoring session; all of the usable history and information of the tutoring session is stored in a packet. As a consequence, the server can “forget” about a client after a packet has been processed.

There are two major reasons for this asynchronous architecture. The first reason appeals to computational efficiency. In an asynchronous model, the server maintains no connection information or memory resources for a particular client. Consequently, memory resource demands do not increase with the number of clients. This process closely resembles the operating system’s notion of a context switch, whereby a server gives each client the impression that he/she is the only user, while simultaneously serving thousands of clients in the background. The second reason appeals to ease of testing. Our asynchronous system can be tested more easily because all of the current state information is explicit in a packet. This greatly facilitates unit testing, debugging, and interoperability. It should be noted that these two reasons for the packet-based asynchronous architecture are entirely computational and practical, as opposed to being motivated by theoretical cognitive mechanisms. It is doubtful that this design decision makes any sense at all from the standpoint of psychological plausibility.

The packet is represented as a binary serialized class known as the *state object.* This class implements an interface for every component of the server. These interfaces have a dual purpose: to declare what data a component needs from other components and to create storage for a component's state for the next turn. Therefore, all of the dependencies between modules are explicit in the state object. The state object further simplifies the system by being the input and output of every component. Therefore, to have a basic grasp of the system, one only needs to understand the dependencies and data in the state object. A packet is essentially a local data structure that permits the server to reconstruct the global history of all modules in the system for that one client.

The components of the system are divided into *modules, utilities,* and the *hub.* The AutoTutor-3D server is a distributed *hub and spokes* application that may reside on multiple servers or on a single server. In the standard hub and spokes configuration, only the hub knows about the existence of the modules. Therefore, any component that provides the data specified by the state object may be interchangeably used. As stated previously, the hub receives a state object from the client and then passes the state object to various *modules* in a scripted order. The basic architecture concept and hub are similar to the DARPA Communicator (Xu & Rudnicky, 2000). However, in AutoTutor-3D, modules are defined by the fact that they input and output state objects, which greatly simplifies communication and interoperability between modules.

Modules may call various utilities, such as LSA facilities or databases that have their own distinctive interface. The AutoTutor-3D architecture is much like a production line in that modules each do a small bit of the work and subsequent modules are dependent on preceding ones. Just as in a production line, the modules are only interested in the work in front of them and forget about previous work. The AutoTutor-3D architecture is somewhat like a blackboard model (Anderson, 1983; Laird, Newell, & Rosenbloom, 1987) in which items are written on a blackboard and modules are only interested in particular items. However, in the present model, the blackboard gets passed from one module to the next, and when the modules are finished with the blackboard they forget about it.

There are theoretical reasons for adopting a modular architecture with packets containing state objects. The architecture emulates a modular cognitive architecture that is also contextually rich. Fodor (1983) and many others in the cognitive sciences believe that the human mind can be segregated into a set of semi-autonomous modules that perform special-purpose functions, such as visual perception, syntactic language parsing, retrieval from episodic memory, executive decisions, and so forth. At the same time, however, the input and constraints operating on each module M should be sufficiently rich and informative about other modules that module M can perform its computations intelligently. Modules that can peak at a limited snapshot of outputs from other modules are inherently limited.

In order to gain a more concrete understanding of our architecture, consider Figure 4. The Client computer sends a state object to the Hub after the student has entered an utterance. The state object is first passed to the Language Analyzer which segments the utterance into main clauses, parses the clauses, and assigns a speech act to each main clause (Olney et al, 2003). The Language Analyzer uses the Conexor EngLite parser (Tapanainen & Järvinen, 1997) as a utility.

The modified state object is sent to the Hub, which then sends it to the Assessor. The Assessor updates the student model and produces a set of predictions about the likely effects of alternative dialogue moves on the student model. The Assessor uses LSA and the Curriculum Script database (i.e., the major content repository of main questions and dialogue moves) to update the student model and make predictions. Next the Assessor passes the updated state object to the Hub, which forwards it to the Dialogue Manager. The Dialogue Manager consults the output of the previous modules as well as the dialogue information state of the previous turn (not shown). The Dialogue Manager subsequently updates the dialogue information state of the state object and provides it with dialogue for the tutor's turn. The updated state object is passed to the Hub, which forwards it back to the client. It should be noted that the state object presented in Figure 4 is highly simplified, but the process of computation is accurately specified.

Insert Figure 4 about here

It is worthwhile to point out a few technical points for those who are interested in implementing similar systems. AutoTutor 3D is written in C# and Visual Basic .NET, both languages that run atop the managed .NET Framework and Common Language Runtime (CLR). The CLR offers language level compatibility (Microsoft .NET Technology Overview 2004). In other words, any component of AutoTutor can easily communicate with any other component, as long as the component is written in a CLR compatible language. Over 30 programming languages have CLR compatible compilers (Richie, 2004), so almost any programmer can use the language of their choice to add functionality to the system. The .NET Framework offers a variety of libraries, including the Remoting system that allows objects to be quickly and transparently accessed over the network and that forms the basis of the scalable infrastructure.

Both the Framework and the CLR virtual machine run on Windows, Linux, Mac OS X, and other operating systems by using open source runtimes such as Mono and Portable.NET.

The AutoTutor-3D server has a variety of protocol translators that convert the data stream from clients into state tables that the modules can understand. These “multi-protocol personal translators” are affectionately known as *muppets*. To make AutoTutor 3D compatible with a new piece of client software, regardless of the language it is written in or the way it transmits data, a programmer simply needs to write a new muppet to parse the data from the client and convert it into a state table. Each muppet can also have its own set of modules and utilities, and a muppet can be instantiated multiple times. This facility permits the construction of multiple virtual AutoTutor servers that work with the same client software, yet differ in functionality. Consequently, experimenters can quickly construct experimental conditions without altering the underlying server code, and can save changes required to the modules to create the variant conditions. Currently, two muppets exist. One muppet reads raw text from a standard TCP/IP connection. The other reads partial state tables constructed by a .NET client that are transmitted through Remoting. An unlimited number of muppets can be constructed as distributed systems on the Internet become progressively more complex.

The standard AutoTutor-3D server we have developed uses four modules and two utilities. As shown in Figure 4, the hub first sends an incoming response, changed into a state table by one muppet, to the Language Analysis module. This module performs surface analyses on the input to determine what sort of speech act the student utterance is. It also performs some text segmentation to aid other modules. The next module in the sequence is the Assessor module. Using LSA and the Curriculum Script, the assessments module determines the quality of the student response with regards to the tutor’s goals for the lesson. It also generates other metrics, such as the length of the response. With this information in hand, the Dialogue Manager module can draw a conversationally and pedagogically appropriate response from the curriculum script. Finally, the state table is passed to the Logger, which records the state of the system for the current turn, before being passed back to the muppet and from there to the client.

The utilities in the standard version of AutoTutor are the Curriculum script and an LSA utility. The curriculum script utility grants modules access to a script of topics and the concepts the tutor is to teach in those topics, tutor moves, correct and erroneous student responses, and common misconceptions on the topic. The LSA utility uses the LSA text similarity algorithm to determine how close a student response is to particular expectations and misconceptions about a topic. The LSA utility was trained on a large corpus of documents that are relevant to the subject matter. For the topic of physics, the corpus was a textbook and 10 articles on Newtonian physics. For the topic of computer literacy, the corpus was a textbook and 30 articles on hardware, the operating system, and the internet.

Dialogue manager

The Dialogue Manager is the architecture's major decision maker. Other modules provide the Dialogue Manager with information it uses to make decisions. One important input is the dialogue information state of the previous turn. This state information is not a history of all previous turns but a summary snapshot of what information is required to successfully continue the conversation. The Dialogue Manager takes in the relevant information and then updates the state object with a new dialogue information state and produces a tutor turn for the client. However, there are multiple steps along the way.

The Dialogue Manager first receives a state object from the hub. This state object is decompressed and translated into an internal format called a *context*. The context contains all dialogue information states and state object elements that are ever used in the Dialogue Manager's computations. In actuality, the Dialogue Manager is a confederation of sub-modules that work on a context. Thus, the context is the common currency for all parts of the Dialogue Manager in the same way that the state object is the common currency for AutoTutor-3D modules.

The planner chain is the first set of sub-modules that the context encounters. In this chain, each planner independently considers the context and suggests plans to move the dialogue forward and successfully tutor the student. Each planner considers a different kind of initiative, including a student initiative, an opportunistic alternative, and a tutor initiative. AutoTutor-3D is a mixed initiative system, which means that the student can ask questions and to some degree take control of the tutoring session. The student initiative planner looks for evidence of student initiative by interpreting student speech acts in the context of the dialogue. When a sufficient student initiative is detected (such as a student question), the student initiative planner pushes a plan tree on the context's plan stack. These plan trees range from question answering sub-dialogues to motivational sub-dialogues to repetitions of previous turns. The opportunistic planner looks for opportunities to clarify or otherwise micro-adapt to the student's progress. For example, the opportunistic planner might push a plan tree to Pump, e.g. "Can you add to that," when the student is doing well but is not verbose. The opportunistic planner looks for instances of the student becoming confused or frustrated with the lack of progress, so the planner pushes a plan tree to let the student how much more information the tutor is looking for. Finally, the tutor initiative planner has inbuilt knowledge of tutoring strategies and what elements a student should cover during the tutoring session. It is the tutoring initiative planner that advances the tutoring session's content in a structured way.

Once the planners have pushed their plan trees, the context is passed to a search module that performs a depth-first search against the plan trees on the plan stack. Each node in a plan tree is visited recursively. When a node in a plan tree is visited, its plan is instantiated. Plans must be instantiated because the plan trees on the plan stack are not specific recipes for action, but rather templates or intentions for action. When the plan is instantiated, a specific update rule is applied to the context, specific dialogue is generated, or both. For example, the Dialogue Manager chooses dialogue moves to maximize student learning. To do this, the Dialogue Manager uses the predictions provided by the Assessor and performs an agent-centered search (Koenig, 1996) akin to MiniMax (Winston, 1984), which is often used by AI programs to play games such as chess. In this instance, however, the Dialogue Manager is not playing against the student, so it tries to maximize the student's score or potential for learning. A specific dialogue move is eventually chosen and added to the context, followed by a data structure in the context being updated to show that the added move has been used. Finally, the plan tree on the plan stack is advanced to the next node. Thus, the results of visitation include the dialogue to be spoken, the updated plans, and the updated context elements.

Plan trees consist of a core set of structural elements which control the flow of the visitation process, along with domain specific plans. This core set includes sequence, iteration, alternation, terminal, and non-terminal elements. Any of these elements can be embedded in another, leading to arbitrarily complex objects. Each of these elements is both a container for plans and a plan itself. As plans, they have termination conditions that mark when the plan is complete. For example, a sequence is completed either when the plans contained within it are exhausted or some success condition is reached which sets the plan complete. The success conditions of plans may be linked together systematically; when one plan is completed, all connected plans are simultaneously satisfied. Completed plans are popped from the plan stack. This process of plan tree traversal continues until a terminal node is visited. The terminal node is a plan to wait for student input. At this point, the context is translated back into a state object, and the state object is passed out of the dialogue manager.

Our Dialogue Manager has similarities to dialogue managers developed by other researchers, but there are some important differences. In systems that implement a general model of collaborative discourse, there is a great emphasis on the negotiation of plans between the agent and the user (Allen et al., 1995). The goals are negotiated as well as the steps or plans to reach them. A collaboration requires that the user and system share the same goals and that they are capable of monitoring progress towards these goals. Nevertheless, we know from research on human tutoring that students rarely take the initiative so it is the tutor that controls the lion's share of the tutoring agenda (Graesser et al., 1995). Students rarely ask information-seeking questions or introduce new topics, for example. Moreover, most students are poor judges of their own progress towards goals (Glenberg, Wilkinson & Epstein, 1982; Weaver, 1990). For example, Person et al. (1994) reported a near zero correlation between students' answering positively to a tutor's comprehension gauging question (e.g., "Do you understand?") and the students' objective understanding of the material. Both Graesser and Person (1994) and Chi et al. (1989) reported a positive correlation between students' answering NO to such comprehension-gauging questions and their objective understanding. The deeper students are better able to identify their knowledge deficits and say they do not understand. Tutoring is a very complex activity in which speech participants have a low amount of shared knowledge and students do not judiciously control their agenda. As a consequence, tutoring is not a conversational context that is or should be collaborative and cooperative in the standard ways that are discussed in the fields of computational linguistics and discourse processing.

In the absence of student initiative, Graesser et al. (1995) found that conventional tutors rely heavily on curriculum scripts (McArthur, Stasz, & Zmuidzinas, 1990; Putnam, 1987) with topics, subtopics, example problems, and topic-specific questions. A pervasive pattern in conventional tutoring is the 5-step dialogue frame (Graesser & Person, 1994; Person, et al. 1995)

1. Tutor asks question
2. Student answers question
3. Tutor gives short feedback on quality of answer
4. Tutor and student collaboratively improve the quality of the answer
5. Tutor assesses student's understanding of the answer

As a reflection of conventional tutoring, AutoTutor-3D's learning goals and the plans to reach them are often set in advance. The Dialogue Manager dynamically constructs very few plans; it dynamically sequences plans based on the current session, but most of those plans are pre-built. In general models of dialogue, the purposes of utterances must be inferred and depend on the task structure. Generating such inferences is manageable in a narrow situational context, such as dialogues between callers and telephone operators or between customers and flight reservation systems. However, there is no successful general dialogue model that has successfully scaled up to handle a large class of contexts or all conversational contexts. In AutoTutor-3D, as with all conversation systems that have been successfully implemented, the purposes of nearly all types of utterances are known a priori. This greatly simplifies the process of discourse interpretation: The tutor has a loosely specified lesson plan whereas student questions are seen as information seeking interruptions to this plan.

Comparisons of AutoTutor with Previous Dialog Managers

AutoTutor-3D plans share similarities with previous work on dialogue modeling in computational linguistics, such as TRINDI (Larsson & Traum, 2000; Traum & Larsson, 2003; The TRINDI Consortium, 2001) and COLLAGEN (Rich & Sidner, 1998; Rich, Sidner, & Lesh, 2001). Historically, different authors have used the term *plan* in a variety of contexts, so it is worth exploring how the AutoTutor-3D notion of plan fits in this history. One distinction that has been made is the distinction between plans and recipes (Lochbaum, 1998). Recipes are merely sequences of actions, whereas plans have beliefs and desires that must be identified and that help drive the actions. The Dialogue Manager of AutoTutor-3D fits the recipe model more than the plan model, although there are vestiges of bona fide plans that are implicitly present (as opposed to being explicitly declared as goals and plans in the computer code).

The objective of the TRINDI project (The TRINDI Consortium, 2001) is to create an approach to dialogue modeling that is suitable for rapid prototyping, promoting domain portability, and accommodating varied theories of dialogue processing. To achieve these three goals, the TRINDI project focuses on information state; the information makes the dialogue distinct and supports the successful continuation of the dialogue. The information state approach is general enough to accommodate dialogue systems from the simplest finite-state script to the most complex Belief-Desire-Intention (BDI) model (Traum & Larsson, 2003; Larsson & Traum, 2000). AutoTutor-3D follows the information state theory of dialogue modeling, which requires:

1. Description of the informational components which constitute the information state.
2. Formal representations of these components.
3. External dialogue which triggers the update of the information state.
4. Internal update rules which select dialogue moves and update the information state.

5. Control strategy for selecting update rules to apply, given a particular information state. AutoTutor-3D uses a context as its model of dialogue information state. Based on a student's utterance, the resulting speech act classification of that utterance, and the current context (as defined in the previous section), the Dialogue Manager updates the context and generates dialogue for the tutor turn. The control strategy consists both of the recursive search over plan trees and the local rules that are applied when a plan is instantiated.

COLLAGEN (Rich & Sidner, 1998; Rich et al., 2001) is based on a theory of discourse structure (Grosz & Sidner, 1986; Lochbaum, 1998) that distinguishes between three kinds of structure: Linguistic, intentional, and attentional. Linguistic structure is the sequence of utterances, whereas intentional structure is the structure of purposes, and attentional state is the focus of attention which records salient elements of the discourse at a particular point. The Dialogue Manager of AutoTutor-3D recognizes linguistic structure (how utterances aggregate into discourse segments) first by interpreting speech acts in the context of the dialogue. Given a speech act, there are 3 possible relations of that act to the current discourse segment purpose (DSP) (Grosz & Sidner, 1986). Either the speech act continues the existing DSP, ends the existing DSP, or begins a new DSP. The Dialogue Manager determines this by looking at both the individual act and its role in the current plan. For example, an information seeking speech act like a student question will start a new DSP, whose purpose is to deliver the sought after information to the student. Non-information seeking contributions will either continue the existing DSP or end the current DSP, depending on whether a plan is satisfied. Shifts in DSP are communicated to the user via discourse markers and canned expressions.

Grosz and Sidner (1986) identify two relations in the intentional structure: Dominance (satisfying X partly contributes to the satisfaction of Y) and satisfaction-precedence (X must be satisfied before Y). The AutoTutor-3D plan stack models dominance and precedence relations. For example, in answering a problem, the student must demonstrate mastery of all elements of the problem. Each of these elements stands in a dominance relationship to the problem; coverage of each contributes to the partial coverage of the problem. Precedence likewise is evident in plan trees for question/answer/clarification dialogues in which the answer must be given before clarification.

Existing implementations of COLLAGEN are still only an approximation of the discourse theory of Grosz and Sidner (1986) because the attentional state is modeled by a stack of plans rather than a stack of focus spaces (Rich et al., 2001). The Dialogue Manager of AutoTutor-3D is similar in this respect: Although the Dialogue Manager does use the context as a cache approach to attentional state (Grosz & Gordon, 1999; Walker, 1996), it also uses a stack of plan trees. The major difference between Collagen plan trees and our Dialogue Manager plan trees is that Collagen plan trees include information about who speaks, the particular conversational participant who is performing what act. This information is wired in more directly in the separate roles and machinery of the tutor versus learner in AutoTutor.

Final Comments

The vision of having a computer communicate with humans in natural language was entertained shortly after the computer was invented. Weizenbaum’s (1966) Eliza program was the first conversation system that was reasonably successful, popular, and widely used. Eliza simulated a Rogerian client-centered psychotherapist. Like a typical Rogerian therapist, Eliza tried to get the patient to do the talking by asking the patient questions about the patient’s verbal contributions. Eliza detected keywords and word combinations that triggered rules, which in turn generated Eliza’s responses. The only intelligence in Eliza was the stimulus-response knowledge captured in production rules that operated on keywords and that performed syntactic transformations. What was so remarkable about Eliza is that one or two hundred simple production rules could very often create an illusion of comprehension, even though Eliza had no depth. It is conceivable that an Eliza with 20,000 well-selected rules might very well exhibit a responsive, intelligent, compassionate therapist, but no one ever tried.

Unfortunately, by the mid-1980's most researchers in cognitive science and artificial intelligence were convinced that the prospect of building a good conversation system was well beyond the horizon. The chief challenges were (a) the inherent complexities of natural language processing, (b) the unconstrained, open-ended nature of world knowledge, and (c) the lack of research on lengthy threads of connected discourse. In retrospect, this extreme pessimism about discourse and natural language technologies was premature. There have been a sufficient number of technical advances in the last decade for researchers to revisit the vision of building dialogue systems. The primary technical breakthroughs came from the fields of computational linguistics, information retrieval, cognitive science, artificial intelligence, and discourse processes. The representation and processing of connected discourse is much less mysterious after two decades of interdisciplinary research in discourse processing (Graesser, Gernsbacher, & Goldman, 2003). The field of computational linguistics has produced an impressive array of lexicons, syntactic parsers, semantic interpretation modules, and dialogue analyzers that are capable of rapidly extracting information from naturalistic text for information retrieval, machine translation, and speech recognition (Allen, 1995; Harabagiu, Maiorano, & Pasca, 2002; Jurafsky & Martin, 2000; Voorhees, 2001). These advancements in computational linguistics represent world knowledge either symbolically, statistically, or a hybrid of these two foundations. For instance, Lenat's CYC system represents a large volume of mundane world knowledge in symbolic forms that can be integrated with a diverse set of processing architectures (Lenat, 1995). The world knowledge contained in an encyclopedia can be represented statistically in high dimensional spaces, such as LSA. An LSA space provides the backbone for statistical metrics that score essays as reliably as experts in English composition (Foltz et al., 2000; Laham, this volume).

Natural language dialogue facilities are not expected to do a reasonable job in all conversational contexts. It depends on the subject matter, the knowledge of the learner, the expected depth of comprehension, and the expected sophistication of the dialogue strategies. We doubt that natural language dialogue facilities will be impressive when the subject matter requires mathematical or analytical precision, when the knowledge level of the user is high, and when the user would like to converse with a humorous, witty, or illuminating partner. A natural language dialogue facility would not be well suited to an eCommerce application that manages precise budgets that a user carefully tracks. Nor would a computerized dialogue system be a good spouse, parent, comedian, or confidant. However, a natural language dialogue facility is feasible in applications that have the following characteristics:

(1) Imprecise verbal content
(2) Low-to-medium user knowledge about a topic
(3) Low-to-medium common ground (shared knowledge) between the user and the system
(4) Earnest literal replies

AutoTutor fits the bill for tutoring students on qualitative domains when the common ground between the tutor and learner is low or moderate rather than high. If the common ground is high, then both dialogue participants (i.e., the computer tutor and the learner) will be expecting a higher level of precision of mutual understanding and therefore will have a higher risk of failing to meet each other’s expectations. We believe that automated tutoring systems with natural language dialogue will be disappointing to the extent that they fail to have one or more of the above four characteristics.

References

Aleven, V., & Koedinger, K. R. (2002). An effective metacognitive strategy: Learning by doing and explaining with a computer-based Cognitive Tutor. *Cognitive Science, 26*, 147-179.

Allen, J. F. (1983). Recognizing intentions from natural language utterances. In M. Brady and R. C. Berwick (Eds.), *Computational Models of Discourse,* (pp. 107-166. Cambridge: MIT Press.

Allen, J.F. (1995). *Natural language understanding*. Redwook City: Benjamin/Cummings.

Allen, J. F., Schubert, L. K., Ferguson, G., Heeman, P., Hwang, C. H., Kato, T., Light, M., Martin, N. G., Miller, B. W., Poesio, M., & Traum, D. R. (1995). The TRAINS project: A case study in defining a conversational planning agent. *Journal of Experimental and Theoretical AI, 7,* 7-48.

Anderson, J.R. (1983). *The architecture of cognition.* Cambridge, MA: Cambridge University Press.

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

Atkinson, R. K. (2002). Optimizing learning from examples using animated pedagogical agents. *Journal of Educational Psychology, 94,* 416-427.

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., Brown, A.L., & Cocking, R.R. (2000)(Eds.). *How people learning: Brain, mind, experience, and school*. Washington, DC: National Academy Press.

Cassell, J., & Thorisson, K. (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., Bassock, M. Lewis, M. W., Reimann, P. and Glaser, R. (1989). Self-explanations: how students study and use examples in learning to solve problems. *Cognitive Science,* **13**, 145-182.

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

Chi, M. T. H., Siler, S. A., Jeong, H., Yamauchi, T. & Hausmann, R. G. (2001). Learning from human tutoring. *Cognitive Science*, *25,* 471-533.

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.

Colby, K. M., Hilf, F. D., & Weber, S.(1971). Artificial paranoia. *Artificial Intelligence, 2,* 1-25.

Collins, A., Brown, J. S., & 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: Erlbaum.

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

Covington, M. (2005). The technological relevance of natural language pragmatics and speech act theory. In C. Forsythe, M.L. Bernard, and T.E. Goldsmith (Eds.), Cognitive systems: Human cognitive models in systems design. Mahwah, NJ: Erlbaum.

Craig, S.D., Graesser, A.c., Sullins, J., & Gholson, B. (in press). Affect and learning: An exploratory look into the role of affect in learning. *Journal of Educational Media*.

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

Fodor, J.A. (1983). *The modularity of mind.* Cambridge, MA: mit Press.

Foltz, P. W., Gilliam, S., & Kendall, S. (2000). Supporting content-based feedback in on-line writing evaluation with LSA. *Interactive Learning Environments, 8*, 111-128.

Forsythe, C., & Xavier, P. (2005). Cognitive models to cognitive systems. In C. Forsythe, M.L. Bernard, and T.E. Goldsmith (Eds.), Cognitive systems: Human cognitive models in systems design. Mahwah, NJ: Erlbaum.

Fox, B. (1993). *The human tutorial dialogue project.* Hillsdale: Erlbaum.

Glenberg, A. M., Wilkinson, A. C., and Epstein, W. (1982). The illusion of knowing: failure in the self-assessment of comprehension. *Memory & Cognition*, **10**, 597-602.

Graesser, A. C., Gernsbacher, M. A., & Goldman, S. (Eds.). (2003). *Handbook of discourse processes*. Mahwah: Erlbaum.

Graesser, A. C., Hu, X., & McNamara, D. S. (in press). Computerized learning environments that incorporate research in discourse psychology, cognitive science, and computational linguistics. In A. F. Healy (Ed.), *Experimental cognitive psychology and its applications: Fetschrift in honor of Lyle Bourne, Walter Kintsch, and Thomas Landauer*. Washington, D.C.: American Psychological Association.

Graesser, A. C., Jackson, G. T., Mathews, E. C., Mitchell, H. H., Olney, A., Ventura, M., et al. (2003). Why/AutoTutor: A test of learning gains from a physics tutor with natural language dialogue. In R. Alterman, & D. Hirsh (Eds.), *Proceedings of the 25th Annual Conference of the Cognitive Science Society* (pp. 1-6). Mahwah: Erlbaum.

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.*

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

Graesser, A. C., & Olde, B. A. (2003). How does one know whether a person understands a device? The quality of the questions the person asks when the device breaks down. *Journal of Educational Psychology, 95,* 524-536.

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

Graesser, A. C., Person, N. K., Harter, D., & Tutoring Research Group (2001). Teaching tactics and dialogue 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-on-one tutoring. *Applied Cognitive Psychology, 9,* 359-387.

Graesser, A C., VanLehn, K., Rose, C., Jordan, P., & Harter, D. (2001). Intelligent tutoring systems with conversational dialogue. *AI Magazine, 22,* 39-51.

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

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.

Grosz, B. J. & Gordon, P. C. (1999). Conceptions of limited attention and discourse focus. *Computational Linguistics*, **25**(4), 617-624.

Grosz, B.J., & Sidner, C.L. (1986). Attention, intentions, and the structure of discourse. *Computational Linguistics*, **12**(3), 175-204.

Harabagiu, S. M., Maiorano, S. J., & Pasca, M. A. (2002). Open-domain question answering techniques. *Natural Language Engineering, 1,* 1-38.

Johnson, W. L., & Rickel, J. W., & 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.

Jurafsky, D., & Martin, J. H. (2000). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Upper Saddle River: Prentice Hall.

Kapoor, A., & Picard, R. (20020. Real-time, fully automated upper facial feature tracking. *Proceedings of the 5th International Conference on Automated Face and Gesture Recognition.*

Kintsch, W. (1998). *Comprehension: A paradigm for cognition*. New York: Cambridge University Press.

Koenig, S. (1996). Agent-centered search: situated search with small look-ahead. In Proceedings of the Thirteenth National Conference on Artificial Intelligence, (pp. 1365). Menlo Park, Calif.: American Association for Artificial Intelligence.

Laham, D. (2005). Emulating human essay scoring with machine learning methods. In C. Forsythe, M.L. Bernard, and T.E. Goldsmith (Eds.), Cognitive systems: Human cognitive models in systems design. Mahwah, NJ: Erlbaum.

Laird, J.E., Newell, A., 7 Rosenbloom, P. (1987). SOAR: An architecture for general intelligence. *Artificial Intelligence, 33*, 1-64.

Landauer, T., & Dumais, S. (1997). An answer to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review, 104*, 2111-240.

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

Larsson, S. & Traum, D. (2000). Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering*, **6**(3-4), 323-340.

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

Lochbaum, K. E. (1998). A collaborative planning model of intentional structure. *Computational Linguistics*, **24**(4), 525-572.

Massaro, D. W., & Cohen, M. M. (1995). Perceiving talking faces. *Current Directions in Psychological Science, 4*, 104-109.

McArthur, D., Stasz, C. & Zmuidzinas, M. (1990). Tutoring techniques in algebra. *Cognition and Instruction*, **7**, 197-244.

Microsoft .NET Technology Overview. (n.d.). Accessed April 15, 2004 at http://msdn.microsoft.com/netframework/technologyinfo/overview/default.aspx

Moore, J. D. (1995). *Participating in explanatory dialogues*. Cambridge: MIT Press.

Moreno, K. N., Klettke, B., Nibbaragandla, K., Graesser, A. C., & the Tutoring Research Group (2002). Perceived characteristics and pedagogical efficacy of animated conversational agents. In S. A. Cerri, G. Gouarderes, & F. Paraguacu (Eds.), *Proceedings of the Sixth International Conference on Intelligent Tutoring Systems*, (pp. 963-971). Berlin: Springer – Verlag.

Moreno, R., Mayer, R. E., Spires, H. A., & Lester, J. C. (2001). The case for social agency in computer-based teaching: Do students learn more deeply when they interact with animated pedagogical agents? *Cognition & Instruction, 19*, 177-213.

Olde, B. A., Franceschetti, D. R., Karnavat, A., Graesser, A. C., & Tutoring Research Group (2002). The right stuff: Do you need to sanitize your corpus when using Latent Semantic Analysis? In W. D. Gray, & C. D. Schunn (Eds.), *Proceedings of the 24th Annual Meeting of the Cognitive Science Society* (pp. 708-713). Mahwah: Erlbaum.

Olney, A., Louwerse, M. M., Mathews, E. C., Marineau, J., Mitchell, H. H., & Graesser, A. C. (2003). Utterance classification in AutoTutor. In J. Burstein & C. Leacock (Eds.), *Building Educational Applications using Natural Language Processing: Proceedings of the Human Language Technology - North American Chapter of the Association for Computational Linguistics Conference 2003 Workshop* (pp. 1-8). Philadelphia: Association for Computational Linguistics.

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

Person, N. K., Graesser, A. C., Kreuz, R. J., Pomeroy, V., & Tutoring Research Group (2001). Simulating human tutor dialogue moves in AutoTutor. *International Journal of Artificial Intelligence in Education, 12*, 23-39.

Person, N. K., Graesser, A. C., & Tutoring Research Group (2002). Human or computer? AutoTutor in a bystander Turing test. In S. A. Cerri, G. Gouarderes, & F. Paraguacu (Eds.), *Proceedings of the Sixth International Conference on Intelligent Tutoring Systems* (pp. 821-830). Berlin: Springer – Verlag.

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**(2), 161-188.

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

Putnam, R. T. (1987). Structuring and adjusting content for students: a live and simulated tutoring of addition. *American Educational Research Journal*, **24,** 12-48.

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

Rich, C., Sidner, C. L., & Lesh, N. (2001). COLLAGEN: Applying collaborative discourse theory to human-computer interaction. *AI Magazine*, **22**(4),15-25.

D.C.: Association for Computational Linguistics

Rickel, J., & Johnson, W.L. (1999). Animated agents for procedural training in virtual reality: Perception, cognition, and motor control. *Applied Artificial Intelligence, 13*, 343-382.

Ritchie, B. (2004) *.NET Languages*. Accessed April 15, 2004 at http://www12.brinkster.com/brianr/languages.aspx.

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

Pasi Tapanainen and Timo Järvinen. (1997). A non-projective dependency parser. In Proceedings of the *5th Conference on Applied Natural Language Processing*, (pp. 64-7). Washington

Traum, D. & Larsson, S. (2003). The Information State Approach to Dialogue Management. In J. van Kuppevelt and R. Smith (Eds.), *Current and New Directions in Discourse and Dialogue*. Dordrecht: Kluwer.

The TRINDI Consortium. (2001). *The TRINDI Book*. Gothenburg: Gothenburg University Department of Linguistics.

VanLehn, K., Graesser, A.C., Jackson, G.T., Jordan, P., Olney, A., & Rosé, C.P. (2004). Natural language tutoring: A comparison of human tutors, computer tutors and text. Submitted to *Cognitive Science*.

VanLehn, K., Jones, R. M., & Chi, M. T. H. (1992). A model of the self- explanation effect. *Journal of the Learning Sciences, 2,* 1-60.

VanLehn, K., Lynch, C., Taylor, L., Weinstein, A., Shelby, R., Schulze, K., et al. (2002). Minimally invasive tutoring of complex physics problem solving. In S. A. Cerri, G. Gouarderes, & F. Paraguacu (Eds.), *Proceedings of the Sixth International Conference on Intelligent Tutoring Systems 2002* (pp. 367-376). Berlin: Springer – Verlag.

Voorhees, E. (2001). The TREC Question Answering Track. *Natural Language Engineering, 7,* 361-378.

Walker, Marylin A. (1996). Limited attention and discourse structure. *Computational Linguistics,* **22**(2), 225-264.

Weaver, C. A. III (1990). *Calibration and assessment of comprehension.* Unpublished doctoral dissertation, University of Colorado, Boulder, CO.

Weizenbaum, J. (1966). ELIZA – A computer program for the study of natural language communication between man and machine. *Communications of the ACM, 9,* 36-45.

Winston, P. H. (1984). *Artificial intelligence.* Reading: Addison-Wesley.

Xu, W. and Rudnicky, A. (2000). Task-based dialog management using an agenda. ANLP/NAACL 2000 Workshop on Conversational Systems, May (pp. 42-47).

Author Notes

The Tutoring Research Group (TRG) is an interdisciplinary research team comprised of approximately 35 researchers from psychology, computer science, physics, and education (visit http://www.autotutor.org). The research on AutoTutor was supported by the National Science Foundation (SBR 9720314, REC 0106965, REC 0126265, ITR 0325428) and the DoD Multidisciplinary University Research Initiative (MURI) administered by ONR under grant N00014-00-1-0600. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DoD, ONR, or NSF. Kurt VanLehn, Carolyn Rose, Pam Jordan, and others at the University of Pittsburgh collaborated with us in preparing AutoTutor materials on conceptual physics.

Table 1

Example AutoTutor-student dialogue with classified dialogue moves of AutoTutor, classified speech acts of the student, and comments.

AutoTutor-1: *The sun exerts a gravitational force on the earth as the earth moves in its orbit around the sun. Does the earth pull equally on the sun? Explain why.* (MAIN QUESTION, This is a main question selected from AutoTutor's currlicum script, which will take approximately 30-100 conversation turns to collaborately answer.)

Student-2: *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-3: *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.)

How does Newton's third law of motion apply to this situation? (HINT. This dialogue move leads the student to articulate a particular sentence, clauses, or proposition.)

Student-4: *What?* (This expression by the student is classified as a Metacommunicative speech act because it addresses the communication channel between AutoTutor and student, in this case a request for AutoTutor to repeat what was last said.)

AutoTutor-5: *Once again, how does Newton's third law of motion apply to this situation?* (REPEAT. AutoTutor repeats its last dialogue move after signalling the function of the present move with the discourse marker "once again.")

Student-6: *Does Newton's law apply to opposite forces?* (This expression by the student is classified as a Verification Question, which calls for a YES/NO answer.)

AutoTutor-7: *Yes, right!* (ANSWER. AutoTutor answers the student's question.)

Hum. (DISCOURSE MARKER. This is a discourse marker that connects different dialogue moves within a turn. Another good discourse marker is "Okay.")

There's still something missing from you explanation. (DISCOURSE MARKER. AutoTutor lets the student know more is expected.)

Newton's Third Law refers to the forces exerted by one body on another ______? (PROMPT. This dialogue move encourages the student to fill in a particular word or phrase of information, in this case the word "body." Prompts are needed for students who are low verbal and to elicit a particular noun, verb, or adjective).

Student-8: *body* (Short Answer)

AutoTutor-9: *You're amazing! Bravo!* (POSITIVE FEEDBACK. AutoTutor gives positive feedback on the answer the student just gave in the previous turn. There are five levels of feedback: Positive, positive neutral, neutral, negative-neutral, and negative).

Let's move on. (DISCOURSE MARKER)

What can you say about the magnitude of the force that the earth exerts on the sun? (HINT)

Student-10: *What is the difference between a force and a vector?* (This expression by the student is classified as a Comparison Question.)

AutoTutor-11: *Interesting question. Perhaps, the following answer helps.* (DISCOURSE MARKER. Notice that AutoTutor hedges on whether the subsequent answer will be satisfying. Hedges are important when a computer lacks confidence.)

Vector is an arrow drawn to scale used to represent a vector quantity. (ANSWER. This answer is fetched from a large textbook on physics rather than being canned. So AutoTutor performs information retrieval.)

Does this answer help you in understanding the material? (ANSWER CONFIRMATION. AutoTutor seeks feedback on the quality of the answer it delivers.)

Student-12: *no.* (Short answer. The student apparently was not satisfied with AutoTutor's answer to the student's question.)

AutoTutor-13: *I might not have understood what you meant.* (DISCOURSE MARKER)

Ask another conceptual physics question if you need more information. (QUESTION SOLICITATION. AutoTutor invites the student to ask a question. Question revisions are common in modern information retrieval facilities.)

Student-14: *What is the difference between force and a vector?* (The student simply repeated the same question, but as shown below, the answer will be different.)

AutoTutor-15: *Force is any influence that tends to accelerate an object; a push or pull; measured in Newtons. Force is a vector quantity and vector is an arrow whose length represents the magnitude of a quantity and whose direction represents the direction of the quantity.* (ANSWER. This answer is once again fetched from the textbook through information retreival methods.)

Okay, I bet you can get this! (DISCOURSE MARKER)

The sun experiences the same magnitude of gravitational pool from the earth, as the earth does from the _____? (PROMP)

Student-16: *sun* (Short answer)

AutoTutor-17: *Yes, right!* (POSITIVE FEEDBACK)

From the sun. (PROMPT COMPLETION. This is the correct answer to the prompt that was delivered in the previous AutoTutor turn.)

Let's keep going. (DISCOURSE MARKER)

What is going on in this situation between the earth and sun? (HINT)

Student-18: *The sun is pulling on the earth because it is bigger.* (This expression by the student is classified as an Assertion because it provides information that helps answer the main question. This answer is identified as a misconception, because both bodies are pulling on each other.)

AutoTutor-19: (No feedback is given to the student even though the student just expressed a misconception. AutoTutor sometimes refrains from negative feedback in order to avoid punishing the student when the student is starting to supply information.)

Let's try this together: (DISCOURSE MARKER)

When two objects of different masses hit each other, which one exerts the greater force on the other, the heavy object, the light object, or are the forces equal? (DIAGNOSTIC QUESTION This question diagnostically verifies that the student really does have a misconception.)

Student-20: *the heavy object.* (Short answer. This is classified as the student's answer to the diagnostic question, which is incorrect.)

AutoTutor-21: *No.* (NEGATIVE FEEDBACK) *the forces are equal in magnitude. Um, well, for every force there is a reaction force, such that the two forces constitute an action reaction force pair. The two forces have opposite direction and equal magnitude. No matter how different the masses of the two objects are, they will exert reaction forces on each other in the same magnitude.* (CORRECTION AutoTutor corrects the student's misconception.)

Table 2

Pedagogical strategies of human tutors and ideal tutors.

(1) *Expectation and misconception tailored dialogue.* The tutor coaches the student to articulate expectations and corrects student misconceptions that are manifested in the dialogue. This is the most common strategy that most human tutors implement.

(2) *Socratic tutoring.* The tutor asks the student illuminating questions that lead the student to discover and correct his or her own knowledge deficits in a self-regulated fashion.

(3) *Modeling-scaffolding-fading.* The tutor first models a desired skill, then has the student perform the skill while the tutor provides feedback and explanation, and finally fades from the process until the student performs the skill all on his or her own.

(4) *Reciprocal teaching.* This is a form of modeling-scaffolding-facing that encourages the student to implement strategies of asking questions, answering self-generated questions, articulating explanations, and summarizing content.

(5) *Frontier learning* and *zone of proximal development.* The tutor selects problems and gives guidance that slightly extends the boundaries of what the student already knows or has mastered.

(6) *Building on prerequisites.* Prerequisite concepts and skills are covered in the session before moving on to more complex problems and tasks that require mastery of the prerequisites.

(7) *Diagnosis and remediation of deep misconceptions.* The tutor diagnoses deep misconceptions that can explain many of the errors and missing information manifested in the student’s performance. These deep misconceptions are corrected after they are diagnosed.

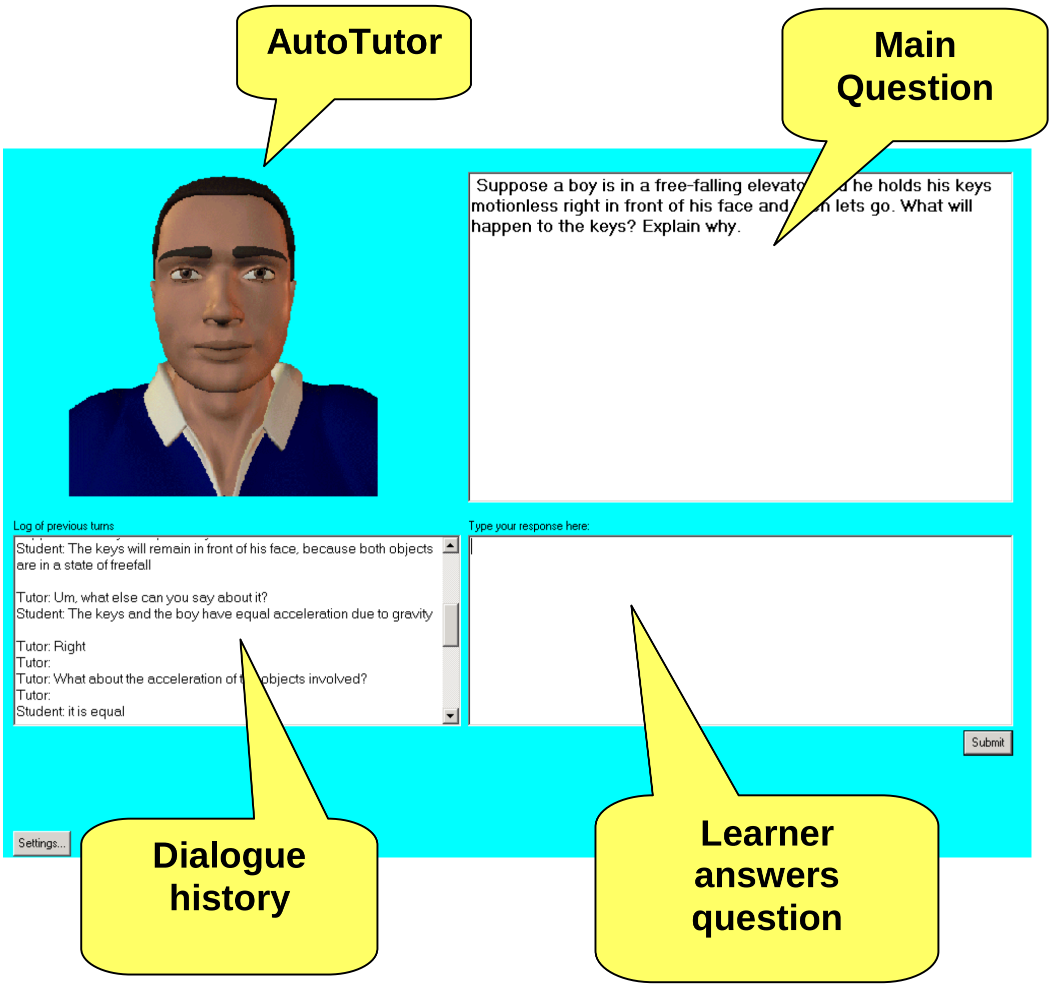
Figure Caption Page

Figure 1: A computer screen of AutoTutor for the subject matter of conceptual physics.

Figure 2. A computer screen of AutoTutor for the subject matter of introductory computer literacy.

Figure 3. A computer screen of AutoTutor on conceptual physics with interactive 3D simulation.

Figure 4. The AT3D Network Architecture. The flow of data proceeds clockwise from the client, returning a response to the client.



Suppose a boy is in a free-falling elevato d he holds his keys motionless right in front of his face and n lets go. What will happen to the keys? Explain why.

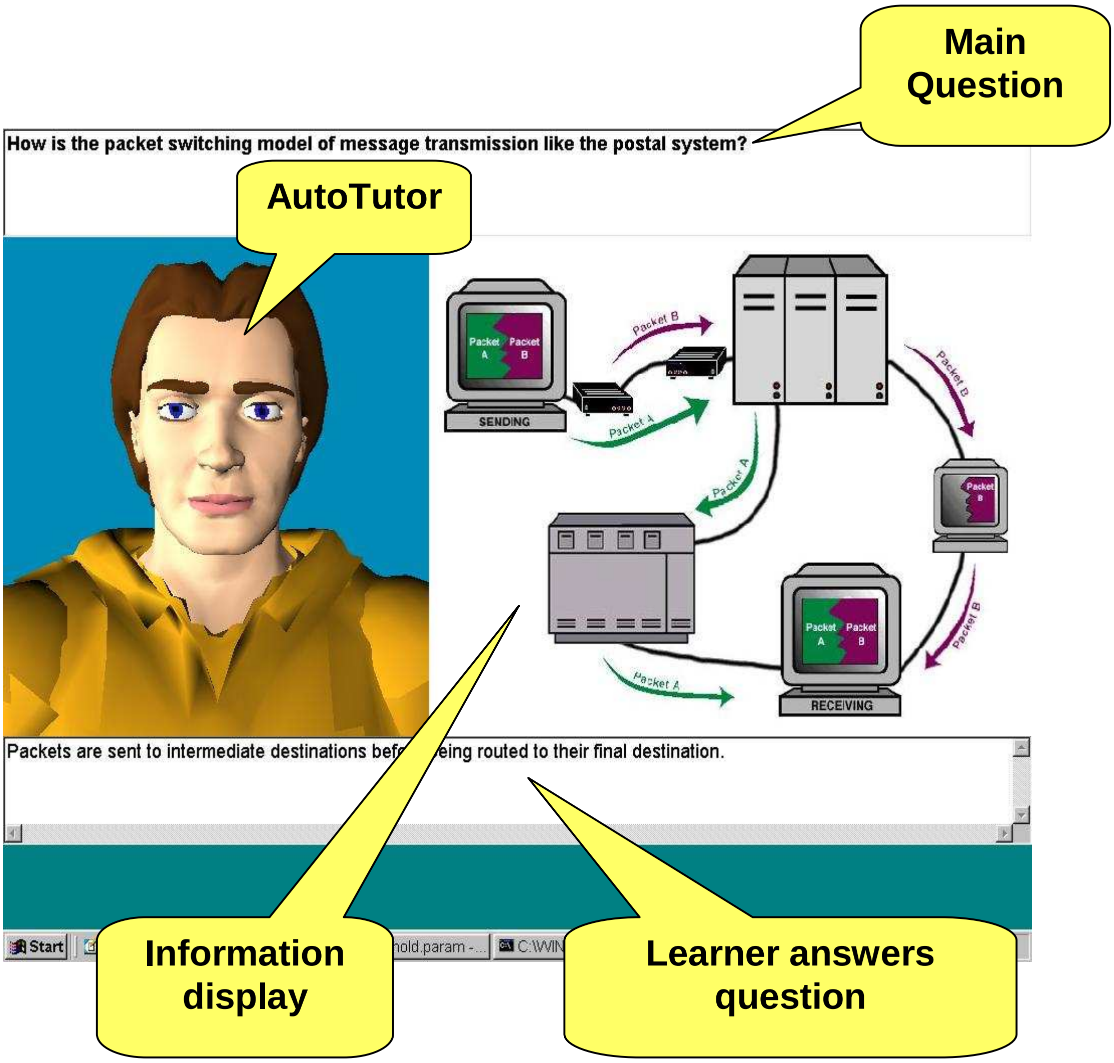
Log of previous turns

Student: The keys will remain in front of his face, because both objects are in a state of freefall

Tutor: Um, what else can you say about it?
Student: The keys and the boy have equal acceleration due to gravity

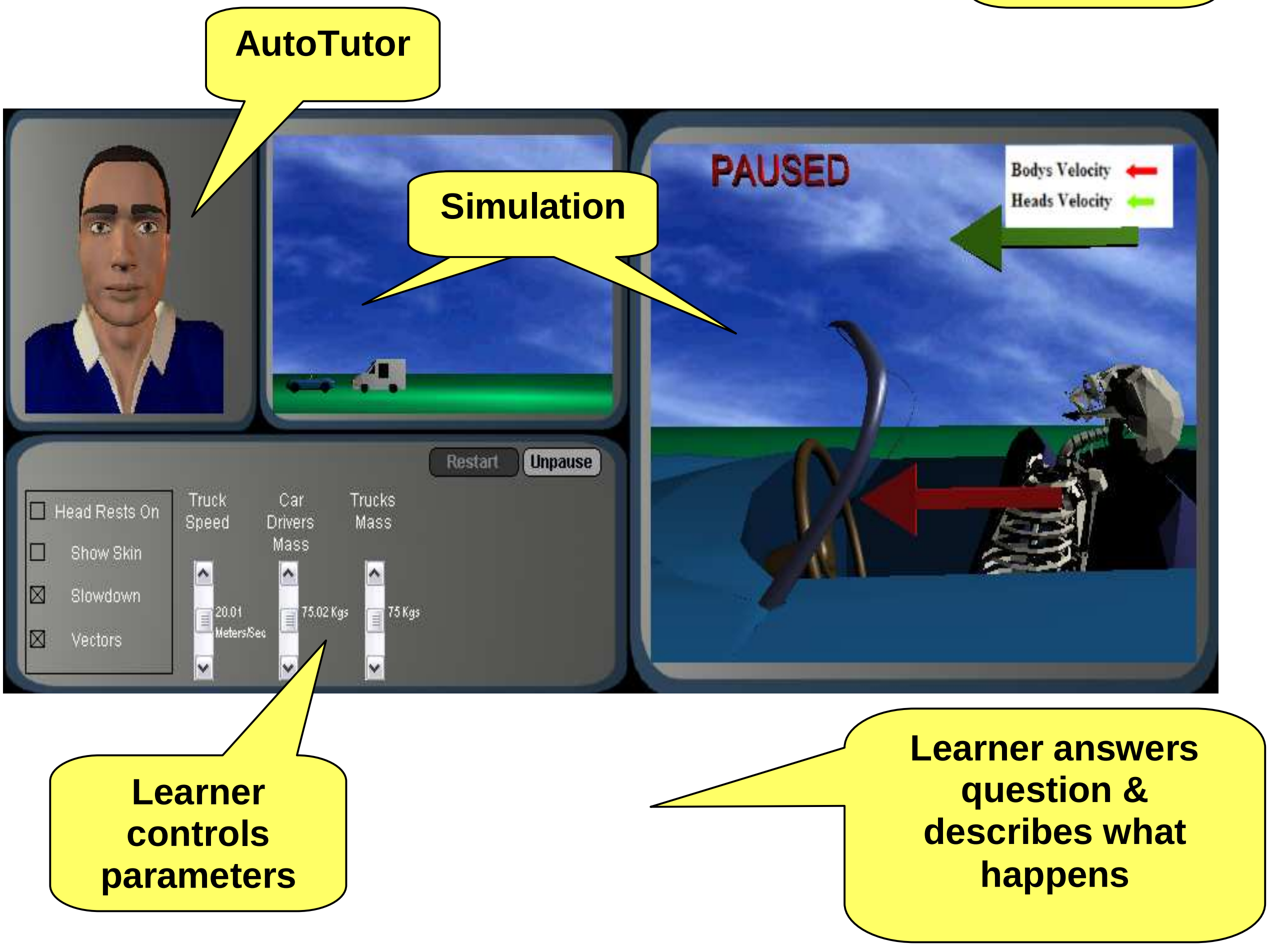
Tutor: Right
Tutor:
Tutor: What about the acceleration of t objects involved?
Tutor:
Student: it is equal

Type your response here:


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

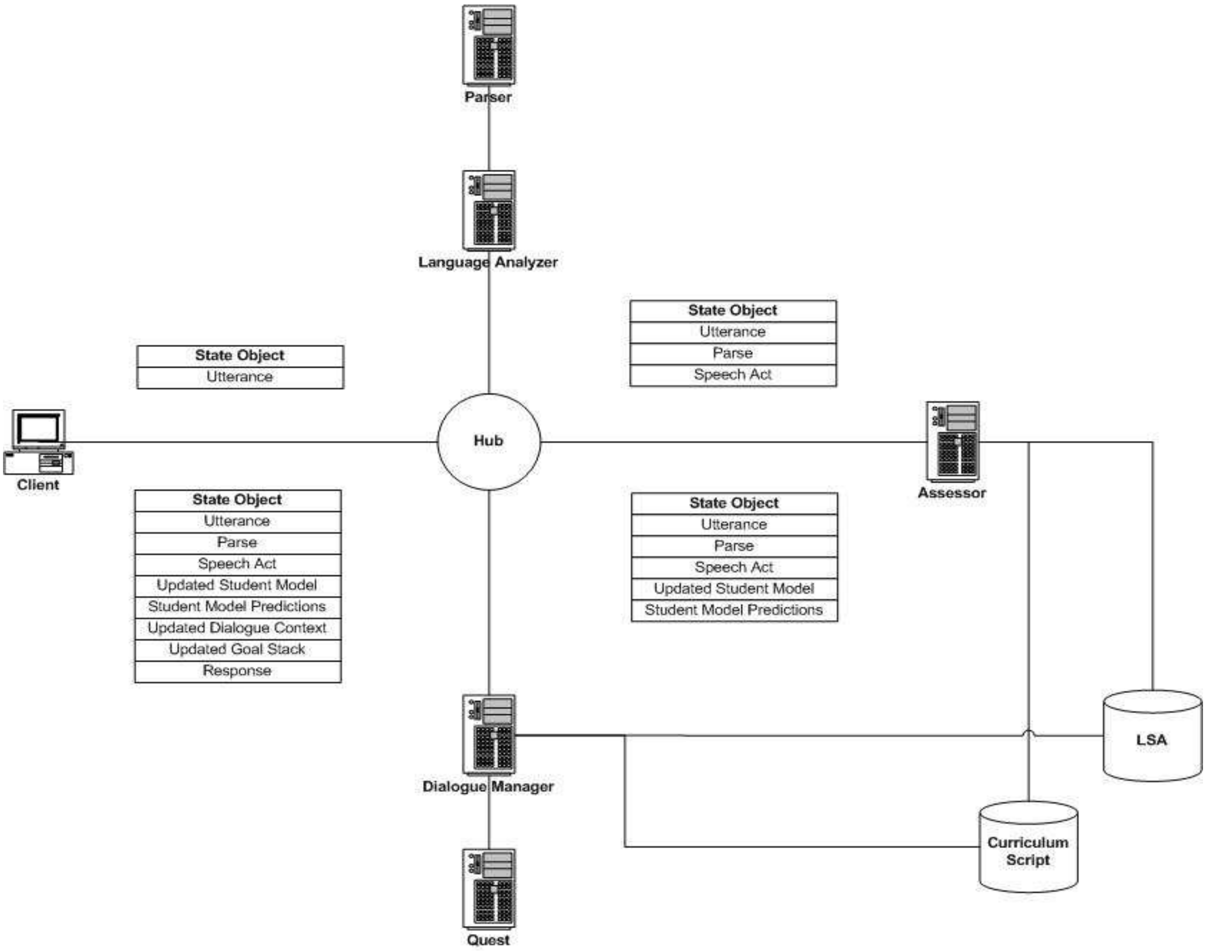
Packets are sent to intermediate destinations bef eing routed to their final destination.

When a car without headrests on the seats is struck from behind, the passengers often suffer neck injuries. Why do passengers get neck injuries in this situation?


Question
AutoTutor
Simulation
PAUSED
Bodys Velocity
Heads Velocity
Restart
Unpause
Learner controls parameters
Learner answers question & describes what happens

☐ Head Rests On
☐ Show Skin
☒ Slowdown
☒ Vectors

Truck Speed: 20.01 Meters/Sec
Car Drivers Mass: 75.02 Kgs
Trucks Mass: 75 Kgs


Parser
Language Analyzer
State Object
Utterance
Parse
Speech Act
State Object
Utterance
Client
Hub
Assessor
State Object
Utterance
Parse
Speech Act
Updated Student Model
Student Model Predictions
Updated Dialogue Context
Updated Goal Stack
Response
State Object
Utterance
Parse
Speech Act
Updated Student Model
Student Model Predictions
Dialogue Manager
LSA
Curriculum Script
Quest