Int J Artif Intell Educ (2014) 24:427–469
DOI 10.1007/s40593-014-0029-5

RESEARCH ARTICLE

# AutoTutor and Family: A Review of 17 Years of Natural Language Tutoring

Benjamin D. Nye · Arthur C. Graesser · Xiangen Hu

Published online: 30 September 2014

**Abstract** AutoTutor is a natural language tutoring system that has produced learning gains across multiple domains (e.g., computer literacy, physics, critical thinking). In this paper, we review the development, key research findings, and systems that have evolved from AutoTutor. First, the rationale for developing AutoTutor is outlined and the advantages of natural language tutoring are presented. Next, we review three central themes in AutoTutor's development: human-inspired tutoring strategies, pedagogical agents, and technologies that support natural-language tutoring. Research on early versions of AutoTutor documented the impact on deep learning by co-constructed explanations, feedback, conversational scaffolding, and subject matter content. Systems that evolved from AutoTutor added additional components that have been evaluated with respect to learning and motivation. The latter findings include the effectiveness of deep reasoning questions for tutoring multiple domains, of adapting to the affect of low-knowledge learners, of content over surface features such as voices and persona of animated agents, and of alternative tutoring strategies such as collaborative lecturing and vicarious tutoring demonstrations. The paper also considers advances in pedagogical agent roles (such as trialogs) and in tutoring technologies, such semantic processing and tutoring delivery platforms. This paper summarizes and integrates significant findings produced by studies using AutoTutor and related systems.

**Keywords** AutoTutor · Intelligent tutoring systems · Natural language processing · Discourse processes · Pedagogical agents · Computer-assisted learning

## Introduction

One grand challenge for education is to scale up the benefits of expert human tutoring for millions of students individually (Bloom 1984). Computer-assisted learning has long been considered as a solution to this challenge, where an automated tutor

B. D. Nye (✉) · A. C. Graesser · X. Hu
Institute of Intelligent Systems, University of Memphis, 410 Fedex Institute of Technology, Memphis, TN 38152-3230, USA
e-mail: benjamin.nye@gmail.com

simulates the pedagogies and conversational patterns of experts. This problem conceptually has a straightforward solution: create a tutoring agent with artificial intelligence that talks with a student, offering the same guidance and support provided by an expert human tutor. Work as early as Carbonell (1970) tried this approach by designing the SCHOLAR tutor to provide Socratic tutoring to learners using natural language text input and output. Despite promising initial results, natural language conversation turned out to be a second grand challenge problem for the AI community. The Turing test, where a computer converses naturally enough to be mistaken for a human, remains a serious challenge in computer science today (Epstein et al. 2009). Decades after the first natural language tutoring system, the vision of a computer tutor emulating all of the capabilities of a human tutor remains a distant goal. However, the good news is that many capabilities of both human tutors and idealized (e.g., theory-based) tutoring strategies can be automated in natural language. Attempts to do this have been pursued by developers of AutoTutor and some of the systems that have evolved after AutoTutor's inception in 1997.

Judicious decisions needed to be made about when to imitate human tutors and when to implement idealized tutoring strategies that a human could not easily implement (Graesser 2011; Graesser et al. 2009). AutoTutor incorporates strategies of human tutors that were identified in human tutoring protocols (Graesser et al. 2009, 1995), as well as ideal strategies derived from fundamental learning research (e.g., modeling-scaffolding-fading, learning progressions), with the basic research goal of determining which of the features help learning and student motivation (Graesser 2011; Graesser et al. 2012a, b). Overall, AutoTutor has been very effective as a learning technology. AutoTutor has produced learning gains that are on average about 0.8σ (standard deviation units) above controls who read static instructional materials (e.g., textbooks) for an equivalent amount of time (Graesser 2011; Graesser et al. 2012b). Learning gains are on par with expert human tutors in computer mediated conversation. On a "bystander Turing test," AutoTutor was indistinguishable from a human tutor when individual conversational turns were evaluated by third-person bystanders who examined transcripts of human-tutor interactions (Person et al. 2002).

This paper tracks three research areas that are central to AutoTutor: human-inspired tutoring strategies, pedagogical agents, and technology that supports natural language tutoring. Some recent papers (e.g., Graesser 2011; Graesser et al. 2012b) have summarized some of the AutoTutor research for less technical audiences, whereas this review offers a broader and more technical perspective on the evolution of each line of research. A major goal of this review is to help the reader understand the similarities, contrasts, and contributions of the systems in the AutoTutor family.

Figure 1 displays a timeline of projects that will be described in this paper, loosely arranged by three foundational lines of research at the top of the diagram. The timeline is organized according to the date that each project was first published. Each of these projects was developed by researchers affiliated with the Institute of Intelligent Systems at University of Memphis, where AutoTutor was developed. Many of these projects were led by different research collaborations, some of which spread across multiple institutions, and represent novel contributions in their own right. Although few of the systems share specific software components or code with AutoTutor, they each inherit theoretical principles, features of AutoTutor's design (e.g., expectation-misconception tailored dialog, which will be described later), natural-language processing algorithms,

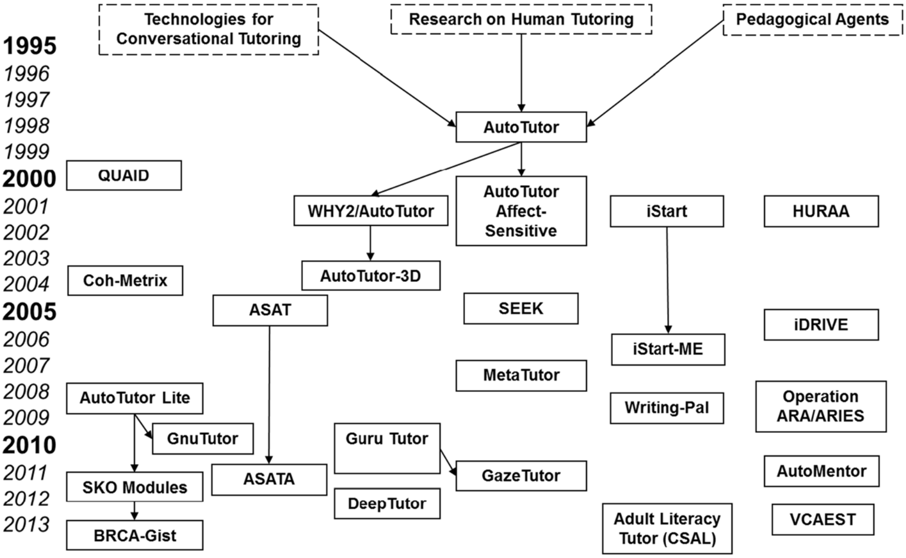


**Fig. 1** Timeline for AutoTutor Family of Projects

and conversational agents. In this sense, these systems form a "family" of related systems that has evolved over time. Due to space limitations, acronyms are used in Fig. 1. Appendix 1 contains a glossary with a short description of each project.

AutoTutor and related systems in the family have tutored computer literacy (Graesser et al. 2004a), conceptual physics (Graesser et al. 2003a; Rus et al. 2013c; VanLehn et al. 2007), biology (Olney et al. 2012), critical thinking (Halpern et al. 2012; Hu and Graesser 2004; Millis et al. 2011), and other topics. AutoTutor approaches have also been extended to push the boundaries of the tutoring interaction, examining the impact of incorporating affect (AutoTutor-AS; D'Mello and Graesser 2012a), gaze focus (GazeTutor; D'Mello et al. 2012), metacognitive skills (MetaTutor; Azevedo et al. 2010), and 3D simulations (AutoTutor-3D; Graesser et al. 2005a). Work has also been conducted to expand the accessibility of AutoTutor, such as a simplified open source release (GnuTutor; Olney 2009), a constrained version designed for web-based authoring and delivery (AutoTutor Lite; Hu et al. 2009), and a framework for sharable web-based tutoring objects (Sharable Knowledge Objects; Nye 2013; Wolfe et al. 2012).

In this paper, we first describe the rationale for natural language tutoring. Subsequent major sections center on the three foundational topics addressed by AutoTutor and later systems connected to the AutoTutor family: 1) human-inspired tutoring strategies, 2) pedagogical agents, and 3) technology that supports natural language tutoring. Key empirical findings on these themes and their contributions are summarized in the Discussion section, focusing on learning gains, affect, metacognition, modality effects, and the roles of pedagogical agents. Finally, the paper concludes with a discussion of the future directions, limitations, and implications of this work for dialog-based tutors and learning technologies in general.

## The Rationale for AutoTutor: Tutoring Through Discourse

Human tutors communicate with the student through discourse, which includes natural language verbal messages, gestures, signals, and non-verbal communication. AutoTutor primarily communicates with learners using an animated talking head that uses natural language (voice or text) and allows unconstrained natural language responses from users. Other successful tutoring systems, such as Cognitive Tutor (Aleven et al. 2009; Ritter et al. 2007) and Andes (VanLehn et al. 2010), organize their interactions around problem-solving interactions without discourse. This raises the question of why one would develop an intelligent tutoring system around discourse rather than circumventing the difficulties of natural language.

We would argue that the important question is not “Why use tutoring discourse?” but instead “How and when should we use tutoring discourse?” Discourse with the student opens up a range of new learning activities related to expressing and communicating knowledge (e.g., self-reflection, answering deep questions, generating questions, resolving conflicting statements). The strengths of natural language tutoring complement a wide range of domains, including traditional problem-solving. From the standpoint of Bloom’s traditional taxonomy of cognitive objectives (Bloom 1956), these activities emphasize comprehending, analyzing, synthesizing, and evaluating knowledge (Anderson and Krathwohl 2001). Dialog-based tutoring helps integrate concepts and problem-solving with domain principles, scaffold domain-specific language, implement abstract strategies, and generate qualitative inferences (Graesser et al. 2001b). Dialog-based tutoring is less effective for remembering shallow didactic facts (Graesser et al. 2004b; Person et al. 2003). Dialog-based tutoring also cannot completely address certain skills and procedures (e.g., no amount of talking about playing piano will make you a pianist if you cannot also practice). As such, natural language tutoring and other types of ITS offer somewhat different pedagogical experiences.

Both natural language and problem-solving tutoring systems support well-validated instructional principles such as timely feedback (Pashler et al. 2005; Shute 2008), active engagement in learning (Prince 2004), difficulty of materials in the zone of proximal development (Metcalfe and Kornell 2005; Wood and Wood 1996), and taking advantage of “teachable moments” when a student experiences an impasse or cognitive disequilibrium (Schwartz and Bransford 1998; VanLehn et al. 2003). However, there are other well-supported instructional strategies that are naturally and distinctively incorporated into dialog. For example, asking context-sensitive deep reasoning questions (e.g., why, how, what if) is one effective strategy in human tutoring (Graesser and Person 1994; Graesser et al. 1995). Similarly, natural language helps a system harness self-explanations (Chi et al. 1994), collaborative interaction (Chi et al. 2008; VanLehn et al. 2007), fostering common ground and terminology (Graesser et al. 2009), and cognitive flexibility through alternative viewpoints (Dillenbourg and Traum 2006; Rouet 2006). Natural language tutors arguably afford a wider range of tutoring interactions, so they are routinely considered in ITS research for tutoring new skills and exploring the rich set of tutoring strategies inspired by human discourse.

## Human-Inspired Tutoring Strategies

Early intelligent tutoring systems had their foundations in expert systems and general cognitive principles rather than tutoring principles specifically. Critiques of these early ITS cited limited attention to modeling human tutoring behavior (Nwana 1990; Self 1990). More attention was placed on rigorously modeling human tutoring behaviors in the early 1990's. For example, Graesser and Person examined over 100 hours of human tutoring interactions (Graesser and Person 1994; Graesser et al. 1995) and Merrill et al. (1992) directly compared human tutoring against intelligent tutoring systems. Given meta-analyses that showed human tutoring is effective (e.g., Cohen et al. 1982), there were two central questions: What features of human tutoring help students learn, and how can we implement these strategies in a computer tutor?

Research on the behaviors of human tutors ultimately led to the design of AutoTutor, including work on collaborative tutoring dialogs (Graesser et al. 1995), pragmatic factors in communication (Person et al. 1995), and strategies to infer student knowledge from question and answer dialog (Person et al. 1994). However, this research also went beyond these mechanisms native to human tutoring and explored more ideal strategies that could be woven into the tutorial interaction. Tutoring strategies included extensive use of deep-reasoning questions and their answers (AutoTutor and later iDRIVE), collaborative lectures (Guru), learning progressions (DeepTutor), reading comprehension strategies (iSTART, iSTART-ME, CSAL), writing strategies (Writing-Pal), affect and engagement detection (AutoTutor Affect-Sensitive, Supportive AutoTutor, GazeTutor), and metacognitive skills such as critical thinking (SEEK, Operation ARIES) and self-regulated learning (MetaTutor). While some of these strategies are descriptive (i.e., observed in human tutoring sessions), many are not consistently or rigorously applied by human tutors, but would be advocated according to ideal principles of learning and pedagogy.

Compared to other natural language tutoring systems, the AutoTutor family has distinguished itself by its strategies for helping students elaborate ideal answers and by the breadth of domains it has tutored. These two factors are probably connected. AutoTutor sessions primarily help students generate correct explanations that solve a problem, while remedying students' misconceptions is a secondary focus. Other successful dialog systems, such as WHY2/Atlas and BEETLE (Graesser et al. 2001b; Dzikovska et al. 2013), have emphasized targeting and repairing misconceptions. While both approaches are effective, misconceptions tend to be highly domain-dependent and are hard for experts to predict. AutoTutor's emphasis on helping students build ideal answers may explain why it has been able to transition to a variety of domains, while the majority of natural language tutors focus on a single domain. This approach limits natural language processing (NLP) to the specific topics for a tutoring session, rather than relying on deep domain-specific NLP that cuts across all dialogs. This strategy was inspired by human tutors, as will be discussed next.

### AutoTutor's 5-Step Tutoring Frame for Collaborative Reasoning

As mentioned earlier, a fundamental question that was: what patterns do typical human tutors follow when tutoring? The discourse patterns of the earliest AutoTutor were inspired by analyses of approximately 100 hours of non-expert human tutoring interactions (Graesser et al. 1995). One corpus included dialogs of graduate students tutoring undergraduate college students on research methods and statistics. A second corpus documented high school students tutoring seventh-grade students on algebra concepts. Non-expert tutors were studied because they perform the bulk of tutoring even though they have moderate domain knowledge and have minimal training on tutoring pedagogies. Moreover, an earlier meta-analysis showed that tutor training did not significantly change learning outcomes (Cohen et al. 1982). Despite having minimal training, non-expert tutors are effective. This finding suggested that non-expert tutors use simple but effective tutoring strategies.

Graesser et al. (1995) reported that non-expert tutors used a limited repertoire of strategies that were guided by an approximate assessment of a students' knowledge. Tutoring sessions showed little active student learning (i.e., students picking the topics), advanced strategies (e.g., Socratic Method, faded scaffolding, modeling), truly anchored learning (i.e., real-life examples), convergence to shared meaning (i.e., precise models of others' knowledge), targeted error remediation (i.e., detecting misconceptions), or affective adaptation. Tutors mainly followed a curriculum script with a set of topics from a chapter and a small set of abstract anchors (e.g., a problem from a textbook) as a focal point for collaborative reasoning, explanations, and construction of meaning.

Collaborative reasoning progressed in a 5-step tutoring frame: (1) Tutor poses a question/problem, (2) Student attempts to answer, (3) Tutor provides brief evaluation as feedback, (4) Collaborative interaction to improve the answer, and (5) Tutor checks if student understands (Graesser et al. 1995). Question asking and answering was a key feature of tutoring. Compared to classroom instruction, students asked many more questions (~26/h vs. 0.1/h) and were asked more questions (104/h vs. 69/h; Graesser and Person 1994). Deep questions and explanations are associated with better student performance. Second, while tutors seldom explicitly pointed out specific misconceptions expressed by the student, they almost always reacted to such errors indirectly. These indirect responses included hints, leading questions to get the student to articulate a correct piece of an answer, or simply asserting a correct answer. These strategies were observed across two separate age groups and domains (Algebra and Research Methods), so they appear to be domain-neutral.

Overall, the tutors "helped students explain good answers" rather than "identifying and correcting misconceptions" (other than simple bugs and slips). This research produced three central tenets to AutoTutor's pedagogical approach at a macro-level:

1. Help students construct explanations of material, such as answers to questions and solutions to challenging problems,
2. Ask questions that tap deep levels of reasoning and that involve collaboration, and
3. Solve problems that involve deep reasoning.

It should be noted that human tutors in Graesser et al. (1995) were not particularly adept at the selection of questions and problems at a macro-level that was tailored to particular students' level of mastery. They tended to ask the same questions associated with a lesson to many students in a scripted non-adaptive fashion. The tutors were much more micro-adaptive but not macro-adaptive, although their micro-adaptivity also used a limited set of strategies.

These strategies are adaptive for non-expert tutors because they can interactively scaffold explanations and solutions to problems despite a very limited understanding of the student's mental model. Since an ITS shares many of these limitations for establishing common ground and knowledge with students, these strategies offered a good starting point for AutoTutor. These principles were the basis for the expectation-coverage strategy, as well as providing design principles for delivering tutoring discourse moves. AutoTutor also included misconception-detection and correction even though student misconceptions are not reliably detected by novice tutors. In practice, finding a good coverage for misconceptions often requires multiple design iterations because the space of possible misunderstandings is vast and even experts have trouble anticipating or diagnosing students' misconceptions. Together, the expectation-coverage and misconception-correction strategies form an *expectation-misconception* discourse framework, which will be described in detail later. Around the same time that this strategy was developed, latent semantic analysis (LSA) emerged in natural language processing (Deerwester et al. 1990; Landauer et al. 1998). LSA, which will be described later, is a statistical method that can compute the similarity of a student statement against various expectations and misconceptions for the concepts being tutored. The convergence of the 5-step tutoring frame, semantic analysis tools such as LSA, and the increasing functionality of animated talking agents provided the three pillars supporting early AutoTutor.

Early versions of AutoTutor attempted to simulate the *discourse moves* of human tutors (e.g., sentences, verbal responses, and non-verbal responses) in a manner that is sensitive to the system's estimates of what the student knows. AutoTutor focuses on the subject matter content of dialogs and the information transferred between the tutor agent and the human student. Domain-relevant dialog is the primary method of conveying information in the system. However, as discussed later, extensions to AutoTutor have added capabilities to detect and transmit nonverbal information such as affective or metacognitive cues.

To help ground the discussion, a physics dialog between AutoTutor and a learner is presented in Table 1. The dialog is annotated to note the discourse moves used by the tutoring agent on each step. In this dialog, AutoTutor is working with a learner to help them explain that two objects in free fall will have the same acceleration. The student initially explains the concept partially, but does not phrase it using physics terms, so the tutor pumps for more information. In response to the pump, the student responds with a related correct statement but does not address the acceleration. This leads the tutor to provide a hint to get the student to talk about acceleration. In response, the student asks a definitional question about the meaning of acceleration that the tutor answers. After this, the tutor repeats the question and the student provides a good answer. Since the student has not stated the direction of the acceleration, the tutor provides a prompt to help the student articulate that the force of gravity acts downward on the objects. If a student cannot cover an expectation despite many hints and prompts, the system will use an assertion to explain the right answer.

AutoTutor relies on a set of distinct discourse moves, with key moves described in Table 2 (Graesser 2011). These dialog moves are driven by an *expectation-misconception* tailored discourse framework. The tutoring system stores a set of anticipated correct answers (expectations) and a set of invalid answers frequently expressed by students (misconceptions). Semantic analysis tools evaluate how well student input

**Table 1** Sample discourse on conceptual physics

**AutoTutor <*Main Question*>:** Suppose a boy is in a free-falling elevator and he holds his keys motionless in front of his him and he lets go. What will happen to the keys? Explain why.

**Student:** The keys would not move but hover at the start point.

**AutoTutor <*Pump*>:** I bet you can explain this a little bit more.

**Student:** The velocity of the keys is the same due to gravity.

**AutoTutor <*Hint*>:** What about the acceleration of the objects involved?

**Student:** What is acceleration again?

**AutoTutor <Answer>:** Let me try to answer your question. Acceleration is the rate that velocity changes over time, like getting faster or getting slower. Acceleration is a vector quantity, so it needs to have a direction.

**AutoTutor <*Hint*>:** Let's try this again. What about the acceleration of the objects involved?

**Student:** The acceleration of the objects in the elevator are the same as the acceleration of the elevator which is the earth's gravitational acceleration.

**AutoTutor <*Short Feedback*>:** Good job!

**AutoTutor <*Prompt*>:** Try this. What direction are the objects going? The objects are falling <hand gesture to cue the student to express something>?

**Student:** downward

**AutoTutor <*Short Feedback*>:** Right.

**AutoTutor <*Prompt-completion*>:** The objects are falling downward.

covers these expectations and detects their misconceptions so that the system can respond with appropriate discourse moves. Semantic *pattern matching* algorithms determine the match between student input and the expectations or misconceptions, whereas semantic *pattern completion* mechanisms generate the AutoTutor's discourse moves to guide students to fill in missing information. User input can be compared to expectations and misconceptions through a variety of semantic analysis techniques, including latent semantic analysis (LSA), regular expressions, word-frequency overlap techniques, and logical entailment (Cai et al. 2011; Graesser et al. 2007b; Rus et al. 2008).

As shown in Table 2, the core discourse moves fill two high-level roles of tutoring discourse: eliciting information from the student and delivering information to the student. A main question initiates each dialog. This is the most superordinate level of the dialog, which encourages deep reasoning to address a fundamental question. AutoTutor dialogs traditionally start with open-ended questions, which have 2–10 expectations (good responses) and 0 to 5 misconceptions (responses that require remediation). Each expectation covers a distinct facet of the ideal answer to the main question. Pumps, hints, and prompts encourage the student to provide more information about each expectation/misconception, with varying degrees of context to lead the student to express increasingly specific information. Short feedback, assertions, corrections, answers, and summaries provide information to students, allowing AutoTutor to correct erroneous statements and review concepts with students who are struggling to produce relevant answers. Each of these discourse moves serves a unique role within the tutoring interaction and is tied to specific dialog and tutoring agent actions (e.g., gestures, facial expressions, calling up diagrams) to help students master concepts from the domain.

**Table 2** Autotutor discourse moves

| Move Type | Description | Example(s) |
|---|---|---|
| Main Question | A question that starts off the dialog, focused on a particular topic or goal | "If the man drops his keys just as the elevator falls, how do the objects move relative to each other? Explain why." |
| Pump | Asking the student to provide more information. | "Anything else?" |
| Hint | Leading question or statement that attempt to get the user to direct the user to answering the main question. | "What do you think about the gravitational force on this object?" |
| Prompt | Leading the student to express a missing word from an important idea for the main question. | "The force on the objects from gravity acts in which direction?" |
| Short Feedback | Signaling about the quality of the student's last statement. | "Great!" (Positive)<br>"Okay." (Neutral)<br>"Not quite." (Negative) |
| Correction | Correcting a misconception or incorrect statement by the learner. | "No, the force of gravity on both objects is equal." (After student claims one is greater) |
| Assertion | Presenting an important idea within the problem or the answer to the problem. | "The force of gravity on both objects is equal." |
| Answer | Response to a learner's question about the definition of a concept. | "A vector is a quantity with both a magnitude and a direction." (In response to "What is a vector?") |
| Summary | Presents the full answer to the main question or problem. | "The magnitude of the force of gravity on each object is equal and all force vectors point down, so…" |

For a full physics problem similar to Table 1 that has many expectations, the dialog can last up to 30 to 100 turns before all expectations are fully covered (Graesser et al. 2005b). However, more knowledgeable students are likely to cover expectations in fewer turns by producing responses with higher relevance. Fewer turns occur for high-knowledge students because AutoTutor tracks a student's overall contributions for each dialog, which acts as a conversation-specific model of which expectations the student knew. More recent systems have also used this information as part of more advanced student models, such as models that estimate affective states (D'Mello and Graesser 2010) and persistent student models (Nye et al. 2014b). Using tutoring dialogs based on this general form, AutoTutor variants have been developed to teach concepts from a variety of domains, primarily in science, technology, and more recently mathematics.

Across the AutoTutor family, most tutoring dialogs have used variations on this 5-step frame and set of speech acts. The earliest versions of AutoTutor followed this pattern closely, while later systems have paired it with complementary strategies, such as vicarious learning or teachable agents (Craig et al. 2006; Millis et al. 2011). While a few systems, such as Guru (Olney et al. 2012), follow somewhat different patterns, the

5-step frame remains a common tutoring mode. Of the five steps, the last (asking if the student understands) is the most commonly omitted or replaced with more effective evaluations of student understanding (e.g., knowledge-check questions). This is because students' assessment of their understanding is typically poor. Since the AutoTutor's authoring tools allow custom dialog rules, variations on these frames are common, but the overall structure is in use today.

## Deep Reasoning Questions

Within this 5-step frame, the earliest versions of AutoTutor explored questions about the effectiveness of deep reasoning questions and the associated collaborative interactions. A key question was: what levels of knowledge show the most learning gains and which components of the discourse impact those gains? While the 5-step tutoring frame can revolve around other learning activities, AutoTutor focused on scaffolding the student's natural language explanation to a deep question or solution to a problem, which remains a central strategy of later systems. Appendix 2 notes the different dialog patterns used by different tutors discussed in this paper. This section discusses strategies used to help students effectively generate explanations to questions that require deep reasoning. Studies are presented that demonstrate the ability of this strategy to transfer effectively to multiple domains, such as computer literacy and physics.

Deep reasoning questions must be answered using explanations, the latter of which are known to produce learning gains (Chi et al. 1994; McNamara and Magliano 2009). Deep reasoning questions have steps such as tend to start with "Why," "How," or "What if." By comparison, shallower questions such as "What" or "Which one" tend to elicit short answers. A taxonomy of deep, intermediate, and shallow question categories was developed in Graesser and Person (1994). Six types of deep questions are present in the current taxonomy: *antecedents* (Why/how was this caused?), *consequences* (What-if? What-next?), *goal-orientation* (Why would someone do this?), *enablement* (What allows this?), *interpretational* (What could this mean?), and *expectational* (Why didn't this happen?). A typical AutoTutor tutorial dialog starts with a deep question from this taxonomy to promote reasoning about domain content and relationships. Students are seldom able to answer these questions without tutoring support, so each main question typically leads to a long series of dialog turns where pumps, hints, prompts, and assertions are used to help the student type a full explanation for the deep reasoning question.

AutoTutor's first iteration was designed to tutor computer literacy skills, such as the fundamentals of computer hardware, operating systems, and the internet (Graesser et al. 2004b, 1999). A curriculum script was designed for the topic of computer literacy. This script was a systematically organized set of deep questions, concepts, corrections, examples, and question-answer pairs. Each topic included a main question, some basic concepts (e.g., key parts of a computer relevant to the question), relevant expectations, anticipated misconceptions, anticipated definitional questions, and a space of discourse moves needed during the conversation (e.g., corrections for misconceptions, answers for definitional questions, hints, hint completions, prompts, prompt completions, a summary). This script provided the problem-specific content for AutoTutor to expect and deliver to the student during tutoring. In designing a curriculum script, each main question was generated by a curriculum designer and was intended to tap deep systems understanding and causal mechanisms on a topic covered by the textbook chapter.

AutoTutor attempted to get the student to articulate each expectation by expressing pumps (e.g., “what else?”), then a hint, then a prompt, and then simply asserting the answer (a bottom out response); this pump → hint → prompt → assertion cycle stopped as soon as the expectation was covered. A good student could articulate the answer with minimal tutoring guidance (i.e., pump or hint) whereas lower ability students needed prompts for specific words or the tutoring directly asserting the expectation (Jackson and Graesser 2006). Moreover, the hints and prompts were carefully generated to fill in the missing content words in the expectation; this adhered to the pattern completion principle that was discussed earlier.

Different versions of AutoTutor were evaluated, each using a small set of fuzzy production rules to drive dialog. AutoTutor 1.1 moved on from a topic after the student covered the expectation or after the tutor delivered a bottom-out assertion. AutoTutor 2.0 continued with a cycle of hints, prompts, and assertions until the student said the material, making them restate their understanding of even a bottom-out assertion before continuing (Person et al. 2003). Despite these differences, both versions performed comparably to each other and both showed learning gains of approximately 0.5σ over controls who studied relevant textbook chapters for an equivalent amount of time. These effect sizes were highly dependent on the type of evaluation test question, with gains of 0.15σ for shallow questions, 0.28σ for deep questions, and 0.64σ for cloze questions (i.e.., filling in words of an explanation). This foreshadowed a pattern of higher learning gains for deep reasoning with AutoTutor (Graesser et al. 2004b, 2010). For a second type of evaluation, a bystander Turing test was conducted to determine whether outsiders could discriminate if a specific tutoring turn was produced by AutoTutor or a human tutor. Bystanders could not differentiate between AutoTutor and the human tutor (Graesser et al. 2005a).

Additional work implemented idealized strategies based on general learning principles derived from the field of education and controlled laboratory studies. To promote *active learning*, one strategy used two cycles of hints, prompts, and assertions. For example, a low-knowledge student might provide little input for the first cycle, but after the tutor asserted the answer, could generate more in the second cycle. This process was abandoned because learners found the two-cycle process frustrating. A second strategy encouraged students to *generate questions* for the tutor to answer, which correlates with learning and can improve meta-cognitive skills (Rosenshine et al. 1996). Unfortunately, AutoTutor was not capable of answering all student questions so this strategy was not feasible. Established learning principles, such as those described in Graesser (2009), continue to drive AutoTutor strategies, but more subtly: interactions are tuned iteratively, balancing many principles, rather than necessarily trying to maximize a single principle. While lab experiments often detect effects as if they are simple, linear, and additive (e.g., “active learning is good”), in practice these effects can have diminishing gains and compete with other principles (e.g., “active learning vs. engagement”). Later research examined more nuanced variations of tutoring discourse, such as Kopp et al. (2012) investigation of the dosage levels for interactive tutoring, as described later.

Shortly after applications to computer literacy, AutoTutor was applied to the domain of Newtonian physics concepts as part of the WHY2/AutoTutor project (often shortened to WHY/AutoTutor). The WHY2 project was intended as a conceptual successor to the early WHY tutoring architecture for tutoring students on interactions within physical systems (Graesser et al. 2001b; Stevens and Collins 1977). This work was developed as one of two parallel approaches to this problem, the other being the WHY2/Atlas project that added dialog capabilities to the Andes tutoring system (VanLehn et al. 2010). WHY2/AutoTutor and WHY2/Atlas systems differed significantly in how they managed dialog. WHY2/Atlas treated dialogs as a finite state graph and emphasized detection of misconceptions, which would trigger sub-dialogs for further diagnosis and remediation (Freedman et al. 2004; Jordan et al. 2006). WHY2/ Atlas also used a physics-specific hybrid classifier for student statements that included dialog templates for physics formulas (e.g., velocity, acceleration), a semantic grammar, and a Bayesian bag-of-words classifier (e.g., similar to LSA in AutoTutor). By comparison, AutoTutor employed a less structured approach to physics semantic evaluation (e.g., adding some basic negation handling) and relied more on the statistical attributes of LSA in semantic matching to expectations and misconceptions. Despite using significantly different approaches to dialog management and semantic matching, both systems performed comparably.

Systematic assessments of AutoTutor in WHY2 on learning gains have been reported (Graesser et al. 2003a; VanLehn et al. 2007). Learning was measured using tests with 4 essay questions and 40 multiple choice questions, many of which were adapted from the Force Concept Inventory (Hestenes et al. 1992). These studies revealed that both WHY2/AutoTutor and WHY2/Atlas performed comparably to non-expert human tutors and with each other. A second study used a pretest/posttest design to compare WHY2/AutoTutor against reading a textbook for a comparable amount of time and against a control condition with no physics materials. This study found learning gains of 0.61σ for students in the AutoTutor condition over the control posttest mean and 1.22σ over the read-textbook posttest mean. Normalized learning gains [(posttest-pretest)/(1.0 – pretest)] showed that students in the AutoTutor condition gained 0.32σ while students in the read-textbook condition gained only 0.04σ over reading irrelevant information. Overall, this confirmed that AutoTutor performed significantly better than both the read-textbook and the control condition. Follow-up studies have shown even greater advantages of AutoTutor over a read-textbook control (Graesser et al. 2012a, b).

The results presented in this section demonstrate how AutoTutor was able to transition to a second domain, from computer literacy to physics. Additionally, it performed comparably to another dialog-based tutor with physics-specific natural language processing and remediation strategies. This indicates that helping students generate explanations to deep reasoning questions is a powerful, domain-independent tutoring strategy.

## Expert Strategies and Collaborative Lecturing

Later work at Memphis considered the question of whether the patterns of expert tutors show different and potentially more-effective strategies than typical tutors (D'Mello et al. 2010b). Since the effect of tutor expertise on learning outcomes is murky due to the lack of a clear definition for an expert tutor (VanLehn 2011), this research focused on highly-qualified practicing tutors. A dozen expert tutors in the Memphis areas were identified, whose qualifications included teaching licenses, five or more years as a tutor, employment at a tutoring agency, and excellent references. Researchers coded 50 h of expert tutoring dialogs to identify dialog moves and dialog modes (D'Mello et al. 2010b; Olney et al. 2012). Expert tutors sometimes used more flexible and complex tutoring strategies than non-experts, but most often used similar approaches as non-expert tutors. The vast majority of dialog moves were scaffolding (46 % of turns) and mini-lectures (30 % of turns).

While expert tutors are often reported to be more interactive (VanLehn 2011), expert tutors in this study spent large amounts of time giving mini-lectures. Since this was not a head-to-head comparison against novice tutors, the implications are not cut-and-dry, though some other studies have reported that expert tutors provided summaries and procedural (how-to) instruction more than novice tutors (Lu et al. 2007). Analysis of the transitions between dialog modes revealed that experts usually alternated lecturing with scaffolding (i.e., explain first and help the student work on a problem/question second; D'Mello et al. 2010b). Overall, lectures were not monologues but were interactive: shorter explanations punctuated by checks for understanding using metacognitive questions (e.g., "Do you understand?") and short questions/problems to elicit student knowledge. These were described as collaborative lecturing, which is lecturing interleaved with shallow questions (D'Mello et al. 2010a).

To explore its effectiveness as a tutoring style, the Guru tutor implemented collaborative lecturing as its primary interaction style (Olney et al. 2012). Guru used collaborative lecturing and exercises that required the students to generate summaries, complete concept maps, and finish cloze tasks. Student performance on these tasks was used to determine which concepts the tutor should target and to determine when the session is complete. This targeting is part of a larger shift toward mixing tutoring with applying knowledge (e.g., problems, exercises), which provide practice and also diagnose concepts where the student struggles.

Guru covers 120 biology topics from the Tennessee Biology I curriculum. Guru produced strong learning gains, approximately 0.72$\sigma$ on a posttest covering biology concepts versus controls who received classroom instruction only (Olney et al. 2012). Compared to AutoTutor, Guru's collaborative lecturing, concept maps, and targeting dialog based on concept map performance represent significant departures. The relative importance of these different elements on learning gains is still being explored. Research on the impact of concept maps will be described in more detail in the section on "Complementary Media".

One takeaway from this line of research was that collaborative lecturing can be an effective tutoring strategy. Since this strategy can also be implemented in other AutoTutor systems, this research supported an additional interaction style which can complement deep reasoning questions. Since both strategies appear effective for both high and low knowledge students, further research is needed to help determine which traits of the student and domain content determine when each approach is more effective. Teasing out when certain strategies should be preferred requires larger samples of human expert tutoring sessions or comparisons of different ITS conditions.

### Learning Progressions

A more recent research question has investigated the impact of adding macro-adaptivity to AutoTutor-style natural-language tutoring. The original AutoTutor dialogs were micro-adaptive (e.g., adapted within a conversation), but they did not use macro-adaptivity (e.g., selecting different main questions for different students). Macro-adaptivity is considered an ideal strategy because most human tutors do not reliably implement it. While human tutors may try to track student knowledge across many sessions, there is no clear evidence that tutors can maintain accurate learner models (Graesser et al. 2009). Macro-adaptivity is particularly important when tutoring over longer periods and when students start with unequal knowledge. The recent DeepTutor project added a macro-adaptive system based on learning progressions for Newtonian physics (Rus et al. 2013a, c). Learning progressions posit that learners go through a pathway of knowledge states, where each state has distinct patterns of understanding and misconceptions.

DeepTutor integrates verbal contributions and problem-based assessment into the detection of learner states by using conversation and short multiple-choice tests to diagnose learning states (Rus et al. 2013c). The system maps student statements and problem-solving activity to different states of understanding each concept. While AutoTutor considers each expectation and misconception independently, DeepTutor associates responses with a spot on the "path" for mastery. For example, some expectations and misconceptions only occur at high levels of understanding. To improve identification of learner knowledge, DeepTutor has applied more advanced natural language processing techniques that include optimized word-to-word similarity and negation handling (Rus et al. 2013c). A recent small-scale evaluation ($n=30$) of DeepTutor showed pretest-posttest learning gains of 0.79σ for a 1-h training session when macro-adaptivity was included (Rus et al. 2014), compared to very little learning when dialogs were not selected dynamically. If even a fraction of this difference is maintained over longer training intervals, adding macro-adaptivity could be a major advance.

### Reading Comprehension Strategy Tutoring

While much of AutoTutor's family have focused on STEM domains, a parallel question was posed: what tutoring strategies are required to train text comprehension strategies, which differ from problem-solving domains like physics? The design of iSTART (Interactive Strategy Training for Active Reading and Thinking) was contemporary to AutoTutor and focused on improving reading comprehension (McNamara et al. 2006). iSTART employs ensembles of agents for tutoring reading comprehension (McNamara et al. 2007). iSTART discourse differs from AutoTutor, concentrating on strategy training rather than content coverage, though self-explanation still plays a major role. iSTART strategies include *paraphrasing* sentences, *bridging* the current sentence to previous material, *predicting* later content, and *elaborating* on content by connecting it to personal knowledge. Additional semantic analysis, such as textual entailment, is used to determine the quality of paraphrasing. iSTART improved reading comprehension differently for skilled readers than for less-skilled readers, for both college students and middle-schoolers (Magliano et al. 2005; McNamara et al. 2006). Less-skilled readers improved on questions about the content of a single sentence. More skilled readers improved on bridging questions, which link knowledge from two or more sentences.

To support extended practice, a game-based version called iSTART-ME (iSTART – Motivationally Enhanced) was developed (Jackson et al. 2010). While the game-based version produced longer contributions, these contributions were of lower quality during a short intervention and only reached equal quality during a multi-week intervention (Jackson et al. 2011). A third system, Writing-Pal (W-Pal), was designed to tutor writing skills relevant to argument essays (McNamara et al. 2012). Writing-Pal features automated essay assessment that gives students feedback on their writing skills (Roscoe and McNamara 2013).

A second line of reading comprehension research started in 2012. As part of the Center for the Study of Adult Literacy (CSAL) project, a tutoring system for improving reading comprehension has been developed. This intervention is designed to help adults who struggle with reading and comprehension of print media. This tutoring system uses a web service driven by scripts authored by the AutoTutor Authoring Tools (ASAT), which is described in the discussion of authoring tools. Compared to prior systems, CSAL is focusing significantly on connecting dialog with interactive interfaces and multimedia. It is difficult for struggling readers to generate text, so the CSAL project uses web media as a secondary channel to communicate with the learner, such as detecting clicks on an interactive graphic that the student can manipulate.

Overall, this line of research has two significant takeaways. First, natural language tutoring systems for reading comprehension must employ significantly different tutoring strategies. Instead of helping the student explain domain *concepts*, the focus switches to practicing and scaffolding comprehension *skills* that must be measured based on the students' use of natural language. Second, this has implications for natural language processing. At very low levels of comprehension, learners cannot even be expected to type answers. At higher levels, more advanced processing (e.g., entailment) is required to determine if comprehension strategies have been mastered.

## Beyond Domain Information: Affect

Researchers have often hypothesized that emotions are an important dimension of human tutoring (Lepper and Woolverton 2002). This raises a few questions: what emotions are important to help tutors adapt, what data can help reliably detect such emotions, and who benefits from adapting to such emotions? AutoTutor-AS (Affect Sensitive, also known as Emotion Sensitive) incorporated affect detection by monitoring body posture, facial expressions, and discourse to classify student emotional states (D'Mello and Graesser 2010, 2012a; D'Mello et al. 2007). Interestingly, the natural language discourse channel had a large impact on predicting emotions such as frustration, confusion, and engagement. For example, affective states are detected by discourse coherence, matches of student contributions to expectations, verbal fluency, hinting, and tutor feedback. Discourse features offer key information about task performance (e.g., are they failing to match the ideal answer?) and activity level (e.g., are they typing a lot or very little?). Discourse context is important for interpreting physical expressions (e.g., facial, posture) and both channels (discourse and physical) improve detection of affect. AutoTutor-AS refers to the system with affect-detection, which can be used to build tutors with different strategies for dealing with affect, such as the Supportive AutoTutor.

D'Mello and Graesser (2012) compared a Supportive AutoTutor with a standard (neutral) AutoTutor for their impact on learning gains. The Supportive AutoTutor empathized with the learner when there were negative user emotions and attributed such difficulties to the material (as opposed to the learner). The Supportive AutoTutor was more effective for low knowledge students (0.71σ gain on posttest scores) when compared to standard AutoTutor, but such gains were only evident on a second session when student difficulty was more pronounced. High knowledge students did not benefit from affect sensitivity at all and sometimes showed lower learning when given emotional support. A buddy-style Shakeup Tutor that attributed emotions to the learner rather than the material did not show positive benefits for learners (D'Mello and Graesser 2012).

Disengagement behaviors have also been considered. The GazeTutor project extended GuruTutor by using a camera to capture student eye-movement data and using that to allow the tutoring agent to react to student gaze, in particular to detect and react to student disengagement (D'Mello et al. 2012). When students appeared disengaged when it was talking, GazeTutor would react by saying statements like "Please pay attention." This was particularly relevant to Guru, which used interactive lecturing rather than highly interactive tutoring. GazeTutor succeeded at increasing attention to the tutor and resulted in small, but not statistically significant overall learning gains (0.26σ, $p$<.178) and moderate, statistically significant gains on deep learning (0.45σ, $p$<.035).

These lines of research indicated that disengagement and negative emotions, such as frustration, are important for tutor adaptation. D'Mello and Graesser (2012b) used data from AutoTutor-AS to develop a model of learning-relevant emotions which includes engagement, surprise, frustration, delight/achievement, disengagement, confusion, and boredom. This model posited that inducing some short-term confusion could be beneficial for learning, which has been verified experimentally (D'Mello et al. 2014; Lehman et al. 2013) verified. Combined with earlier findings, this indicates that learning is improved when (a) disengagement is reduced (e.g., by reacting to gaze), (b) brief confusion is induced (e.g., by presenting conflicting information), and (c) frustrated learners with low knowledge receive affective support.

## Learning about Thinking: Metacognition

AutoTutor descendants have also been applied to study critical thinking, metacognition, and self-regulated learning. The SEEK (Source, Evidence, Explanation, and Knowledge) web tutor was designed to help college students improve their critical thinking by evaluating the credibility and relevance of information as part of the scientific inquiry process (Graesser et al. 2007c). SEEK did not use a full dialog system, but instead embedded spoken hints and structured note-taking to evaluate the information hit during a constrained web search (seven cached sites). Six primary skills were practiced: (1) Asking deep questions, (2) Collecting information from multiple sources, (3) Evaluating the validity of information, (4) Integrating information from multiple sources, (5) Resolving inconsistencies, and (6) Constructing a causal model of a system. Unfortunately, this tutor did not significantly improve students' skills in finding relevant information. This lack of improvement was mainly attributed to short length of training (1 h), combined with the complexity of each skill involved (e.g., asking deep questions).

Operation ARIES (Acquiring Research, Investigative, and Evaluative Skills) also targeted critical thinking skills in a serious game (Millis et al. 2011). Operation ARA (Acquiring Research Acumen) continued this work by refining and extending ARIES into a learning game for wider distribution by Pearson Education (Halpern et al. 2012). ARIES/ARA address a set of skills more specifically related to scientific inquiry, such as: Theories and hypotheses, Independent and dependent variables, Validity, Replication of results, Experimental controls, Sample size, Experimenter bias, Making causal claims, and Generalizability. One component of ARIES/ARA attempts to get the student to articulate scientific principles in natural language by using the pump-hint-prompt-assertion cycles of AutoTutor. Another pedagogical approach is to present specific research cases that have methodological flaws and to ask students to articulate the flaws in natural language.

While the game has not yet been compared against static materials (e.g., texts), evaluations of Operation ARA reported learning gains of 1.4σ on a test of scientific research skills when compared with controls who received no instruction (Halpern et al. 2012). Compared to SEEK, this success may possibly be attributed to additional time-on-task and a more cohesive set of skills. Students used ARA between 8 and 12 h, an order of magnitude longer. There were also motivational features through game elements, such as a coherent game narrative and Jeopardy-like training rounds. ARA also used much more AutoTutor discourse compared to SEEK, with a mixture of tutoring and vicarious learning (watching agents talk with each other). SEEK relied significantly on supporting self-regulated search, whereas ARA's more structured environment could have also influenced learning gains.

Self-regulated learning skills (SRL) motivated the design of MetaTutor, which used discourse-based agents to support a hypermedia learning environment covering biology topics (Azevedo et al. 2010). Like AutoTutor-AS, a recent version of MetaTutor used real time analysis of facial expressions to classify student emotions during tutoring. Azevedo et al. (2012) reported higher learning efficiency when tutors assisted students with prompts and feedback as they explored the hypermedia pages (0.84σ). However, that learning efficiency score included only time spent interacting with the material and not time interacting with the agents. After considering the entire time spent using the system, learning efficiency was approximately equal. Even if learning efficiency for the domain is unchanged, improved self-regulated learning might offer valuable long-term benefits. Unfortunately, other research groups (e.g., Roll et al. 2011) have likewise failed to show that improving self-regulated help-seeking changes domain-specific learning gains even over multiple-month interventions.

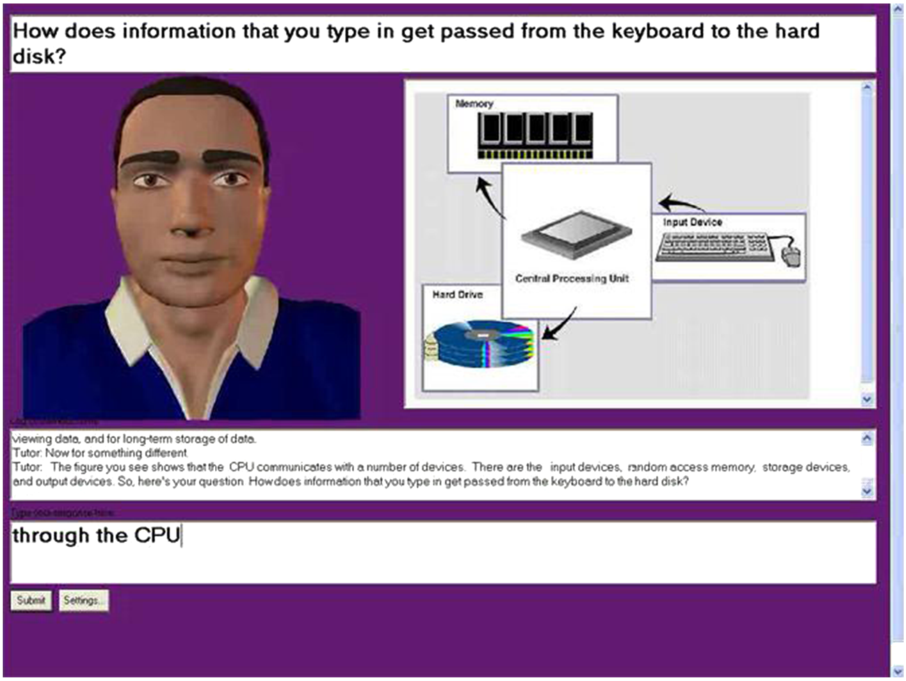
With that said, a few studies have shown that supporting self-assessment (Long and Aleven 2013) and other metacognitive skills (Koedinger et al. 2009) can improve domain learning gains. More research is still needed to definitively show which self-regulated learning skills improve learning in specific domains, as well as metacognitive skills training transfers to new systems and domains. In general, research needs to determine when tutoring SRL outweighs the benefits of spending time directly tutoring domain content.

## Pedagogical Agents and Complementary Media

AutoTutor uses a mixture of media and animated pedagogical agents to deliver tutoring dialogs. While educational software implemented basic animated agents (e.g., sprites) since the start of personal computers, advances in graphics in the mid-1990's made more advanced animated agents feasible. AutoTutor emerged during a wave of research on animated interface agents (Dehn and Van Mulken 2000; Johnson et al. 2000; Nwana 1996), a trend that continued within educational technology and ITS specifically. AutoTutor was part of an early generation of animated agents in tutoring systems and intelligent environments, along with contemporary systems such as STEVE (Johnson and Rickel 1997), Baldi (Massaro 1998), Cosmo (Lester et al. 1998) and the Teachable Agents project (Brophy et al. 1999).

The original AutoTutor relied on a talking head driven by Microsoft Agent that incorporated speech synthesis and supported facial expressions and intonation that were tied to the quality of student contributions (Graesser et al. 1999). Figure 2 shows an example of this early AutoTutor interface. While the quality of animation and speech synthesis has improved drastically since this time, the role of the animated agent has remained fairly consistent up until recently. The tutoring agent in AutoTutor offers a universal interface that students can understand, across a variety of subject matters. Additionally, agent conversations are interactive and capable of offering speech rather than text when this is needed. This line of research has spanned multiple major projects exploring the value and role of pedagogical agents.

Projects have recently shifted from using single-agent interfaces to multiple agents, part of a larger trend in ITS where ensembles of agents are increasingly common



How does information that you type in get passed from the keyboard to the hard disk?

viewing data, and for long-term storage of data.
Tutor: Now for something different
Tutor: The figure you see shows that the CPU communicates with a number of devices. There are the input devices, random access memory, storage devices, and output devices. So, here's your question How does information that you type in get passed from the keyboard to the hard disk?

through the CPU

**Fig. 2** Early AutoTutor Interface

(Graesser et al. 2008, 2014; Halpern et al. 2012; McNamara and Magliano 2009; Millis et al. 2011; Woolf 2009). Table 3 lists common types of roles that pedagogical agents can play in a tutoring system. The roles for pedagogical agents have undergone significant shifts over time. Single non-interactive presenters, opponent agents (e.g., chess opponents), and non-player characters that populate a reactive game world were already common before systems like AutoTutor emerged. While such agents are still used, AutoTutor and similar systems have contributed to the development of a new generation of pedagogical roles, such as vicarious learning presenters (iDRIVE; Craig et al. 2006), affectively supportive agents (Supportive AutoTutor; D'Mello and Graesser 2012a), student peer agents (Operation ARIES; Millis et al. 2011), and agents that enhance virtual worlds (VCAEST; Shubeck et al. 2012). As with all new technologies, there have also been dead ends. This section reviews significant findings related to the features and roles of animated agents, as well as complementary media such as simulations (AutoTutor-3D; Graesser et al. 2005a) and concept maps (Guru; Olney et al. 2012).

## Surface Features: Animation and Speech

A key question for research on animated pedagogical agents has been what features contribute to learning. Studies with AutoTutor have indicated that the content of conversations is more important than the delivery medium or input modality (Nye et al. 2014a). For example, work with AutoTutor showed no statistically significant degradation for lesioned versions that lacked the animated talking head or conversed using text rather than speech (Graesser et al. 2003b). Both factors were secondary to the presenting the right content, since removing the agent animation (no avatar) or presenting print only (no avatar or voice) each resulted in a statistically non-significant −0.13σ reduction in learning gains (Graesser et al. 2003b; Link et al. 2001). The redundancy effect for multimedia learning, which posits that duplicating text and voice

**Table 3** Common animated pedagogical agent roles

| Role | Purpose |
|---|---|
| Presenter | Non-interactive agent that delivers static content. |
| Tutor | Supports learning by asking questions and giving trustworthy information. |
| Peer/Student | Acts as a peer learner and may demonstrate either correct answers or misconceptions. May sometimes be a teachable agent that can learn to perform certain tasks over time. |
| Supportive | Provides affective or motivational feedback (e.g., "Let's keep trying."). |
| Opponent | Competes against the learner on some task or game. |
| Navigational | Helps the learner decide what to do next or locate learning materials |
| Non Player Character (NPC) | Reacts to user behavior as part of an interactive world or simulation. |

hurts learning, was also tested using AutoTutor. Contrary to studies that showed negative effects for redundant voice and text, Graesser et al. (2008) reported a non-significant positive (+0.34σ) impact on learning, rather than a negative effect.

Later research tested the impact of students providing their answers with voice input as compared with typed text. Voice input and typed input performed comparably, with a slight advantage to text input due to errors in voice recognition (D'Mello et al. 2011). Overall, this research strongly indicates that the content, rather than the animation or speech, is primarily responsible for learning gains.

### Vicarious Agent Demonstrations

Another line of research has investigated the question of whether students can learn by watching one agent tutoring another student agent. Craig et al. (2006) developed iDRIVE (Instruction with Deep-Level Reasoning Questions in Vicarious Environments) to study this approach. This project compared the effectiveness of AutoTutor tutoring interactions against a vicarious learning system where a peer student agent asked a series of deep questions and a teacher agent promptly answered each question, with no meaningful input by the user (i.e., vicarious learning by observation). This vicarious learning research followed up on a study that reported that adaptive conversational interaction was only slightly more effective than presenting succinct, targeted script content that directly answered the main question (Graesser et al. 2004b). Research on iDRIVE revealed that vicarious learning with deep questions performed comparably to AutoTutor on physics (Gholson et al. 2009). Vicarious dialogs where the peer student modeled asking deep questions also increased question *asking* by students, which is a metacognitive strategy that improves learning (Craig et al. 2006; Rosenshine et al. 1996).

iDRIVE lead to studies that examined different vicarious self-explanations (Craig et al. 2012). These studies evaluated learning gains under four conditions: a content monolog, questions + answer content responses, "self-explanations" stated by a peer agent, and questions + self-explanations. A study of college students reported that low knowledge students benefited significantly more from the question + explanation condition (34 % learning gain vs. 7 % for high knowledge students; Craig et al. 2012). A follow-up study examined high-school students in different ability tracks (honors vs. standard), but comparable prior knowledge on pre-tests. This study found that students of both honors and standard classes significantly benefited from questions + explanations ($p<0.01$), with honors students showing slightly higher learning gains (Craig et al. 2012). Considering both studies, these findings imply that that low knowledge students, even with different ability levels, benefit from vicarious "self-explanations" that help build a mental model. Craig et al. (2012) hypothesized that high-knowledge students benefit less due to mismatches between the students' existing mental models and the agents' explanations. These results indicate that vicarious deep questioning and explanations, which are significantly simpler to author than dynamic tutoring, can offer significant advantages for students with low knowledge. In particular, vicarious dialogs can model new skills and interactions (e.g., question-asking) that the human learner cannot yet accomplish even with help.

## Trialogs: Tutor, Peer Student, and Human Student

A landmark extension of AutoTutor combined vicarious learning with interactive tutoring in order to take advantage of the complementary strengths of these two approaches. Operation ARIES introduced agent trialogs, where the human student was situated in a three-party conversation between a second agent student and a teacher agent (Millis et al. 2011). This design supports a mixture of vicarious learning and interactive tutoring, allowing the system to combine the benefits of both. Several types of interaction are possible. For students who are having trouble with the material, vicarious learning is suitable, but the human is drawn in periodically by asking them to answer simple yes/no verification questions. For students who have deep mastery of the material, it is appropriate to have teachable-agent designs, with the human student teaching the simulated student, detecting errors in their reasoning, and resolving conflicting opinions between the two agents (Lehman et al. 2013). This design was particularly beneficial for tutoring critical thinking in ARIES, allowing different agents to represent information sources that disagree. Studies indicate that interactive trialogs significantly improve scientific reasoning beyond no-dialog vicarious sessions (Kopp et al. 2012).

## Navigational Agents in Hypermedia

One line of research explored the question of whether conversational agents are effective guides to help learners find and utilize learning resources, HURAA (Human Use Regulatory Affairs Advisor) was a web-based tutor for training ethics in human subjects research, with a curriculum based on United States Federal agency regulations (Hu and Graesser 2004). HURAA was integrated with a learning management system that included hypertext, multimedia, help links, glossaries, links to external sites, case-based reasoning lesson modules, and a conventional e-text presentation of material. AutoTutor's conversational capabilities were applied in ways that differed from other systems. Firstly, the conversational agent was used as a guide to help users navigate the system rather than collaboratively answering deep reasoning questions. Secondly, a question and answer system called Point & Query supported asking questions by means of a context-sensitive dropdown list of relevant questions (Langston and Graesser 1993). Finally, semantic analysis supported users' natural language queries for help on a particular topic.

Evaluations of HURAA reported that users had significantly higher free recall (1.19σ), cued recall (0.56σ), cloze recall (0.58σ), and accuracy for retrieving relevant documents (0.67σ), when compared to controls who used a conventional hypermedia and e-text presentation of the same material (Hu and Graesser 2004). However, HURAA did not change the ability of users to diagnose problematic issues in cases, indicating that the case-based lesson modules were not significantly more effective than simply reading an expert opinion on the case (the control condition). Additionally, a second study reported that using the agent as a navigational guide did not significantly improve any of these measures, nor did it change the users' impressions of the system significantly (Graesser et al. 2003c). As noted earlier, the SEEK agents were also not helpful in improving navigation in a hypermedia system on plate tectonics. In both systems, emphasis was placed on answering students' questions and information retrieval rather than the agent posing deep reasoning questions and collaboratively answering the questions. Given the poor performance of navigational agents in both SEEK and HURAA, this suggests that animated agents may not offer much advantage during information search and retrieval tasks when compared to more traditional text interfaces. Perhaps navigational guides may be more effective when the learner is very lost or frustrated and needs guidance, but this needs further investigation.

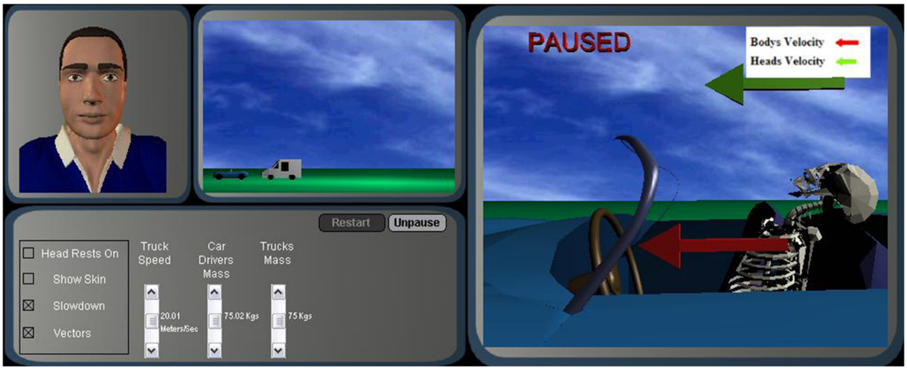
### Integrating Agents into Virtual Worlds

A more recent question has explored the potential benefit of enhancing virtual learning practice environments with deep reasoning questions. VCAEST (Virtual Civilian Aeromedical Evacuation Sustainment Training) trains civilian medical personnel on federal guidelines for triage in emergency situations, particularly those where military aircraft will be used to evacuate victims of an emergency, such as a natural disaster or mass-casualty attack (Shubeck et al. 2012). VCAEST uses AutoTutor Lite, which will be introduced shortly, to work with trainees within a 3D virtual world. This world includes a tutoring agent and an ensemble of in-game non-player character agents, some of whom represent injured individuals who must be stabilized and evacuated. The project represents a mixture of direct intelligent tutoring with an interactive environment populated with task-relevant agents. In VCAEST, natural language tutoring dialogs "pop up" and ask questions about virtual world events (e.g., bad triage decisions). Virtual world events that launch tutoring dialogs is an interesting approach to integrating discourse with authentic scenarios of social action. However, it is pedagogically unclear when to interrupt the simulation with these dialogs and how to manage tutoring if multiple learners are in the world. The VCAEST project is ongoing, so the effectiveness of learning in this tutoring-enhanced virtual world is still being evaluated.

### Complementary Media: Simulations and Concept Maps

AutoTutor and related systems have added value by complementing the tutoring agents with various types of media, ranging from simple images up to simulations. AutoTutor-3D used three-dimensional simulations to help students represent the physics problems they were working on (Graesser et al. 2005a). That is, AutoTutor used an interactive simulation showing the objects and their spatial orientation within the problem. 3D representations and simulations were added because they were believed to promote deeper understanding of physics. In addition to conversing with the AutoTutor agent, students were able to modify parameters of the simulation and launch the simulation to display what will happen. During this process, the tutoring agent was able to request that the students predict what would happen and ask them to compare what actually happened in the simulation with their expectations. However, AutoTutor mainly posed deep questions similar to WHY2/AutoTutor, rather providing simulation-specific scaffolding. An example of this interface is shown in Fig. 3.

Results revealed that most students made little use of the simulation features and had trouble using them productively. There was only a 0.22σ gain for the AutoTutor-3D condition over one with no 3D objects, which did not quite meet significance at $p=0.05$ (Jackson et al. 2006; Kim et al. 2005). The number of times students used the



☐ Head Rests On
☐ Show Skin
☒ Slowdown
☒ Vectors

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

**Fig. 3** AutoTutor-3D Interface

simulations correlated with learning ($r$=.51, $p$<.01), but total usage was low and the overall impact on students was modest. High performing students made greater use of the simulations, however, and appeared to gain some benefit. These results confirmed earlier findings that students have trouble using simulations productively and are unable to systematically manipulate different combinations of parameters, observe what happens, and record the results (Klahr 2002). Students clearly need more guidance to use simulations profitably. Agents might be designed to provide such guidance, but would likely need to be integrated with the simulation more tightly than in AutoTutor-3D.

GuruTutor (see Fig. 4) integrated secondary multimedia (e.g., graphics, video) more tightly than many earlier members of the AutoTutor family of ITS (Olney et al. 2012). The agent strategically points to elements in a display during the course of the tutoring. Guru also uses concept maps to provide demonstrations and students interact with

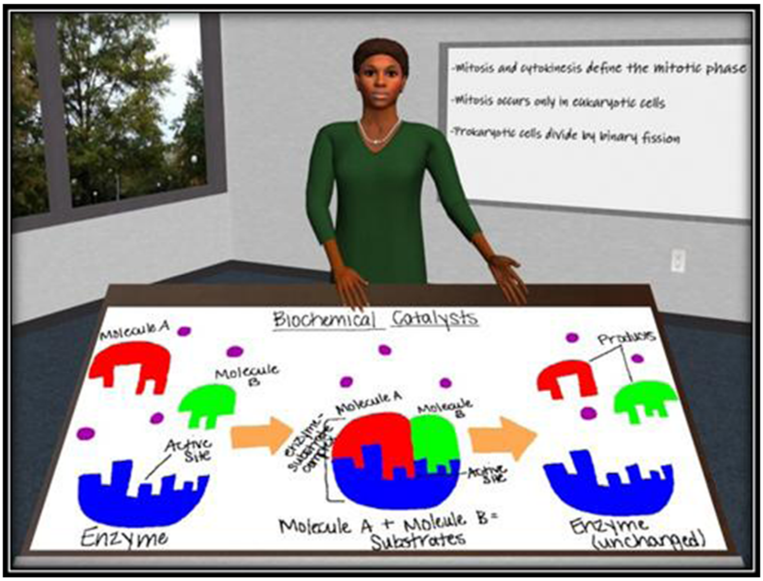

-Mitosis and cytokinesis define the mitotic phase
-Mitosis occurs only in eukaryotic cells
-Prokaryotic cells divide by binary fission
Biochemical Catalysts
Molecule A
Molecule B
Active Site
Enzyme
Enzyme-substrate complex
Molecule A
Molecule B
Active Site
Molecule A + Molecule B = Substrates
Products
Enzyme (unchanged)

**Fig. 4** Guru Biology Tutor Interface

concept maps during exercises. While Nesbitt and Adesope's (2006) meta-analysis reported an effect size of 0.82σ for concept maps, the impact of concept maps in Guru has been inconclusive. Time spent on concept maps and errors on concept map tasks did not show reliable correlations with learning across two different topics (Person et al. 2012). This may be due to the types of content involved : time spent on an easier, more fact-based concept map correlated with learning gains but there was no correlation between time spent on concept maps for a second topic that was more difficult and procedural (Person et al. 2012). As a comparison, Betty's Brain showed consistent learning gains using concept map tasks (Leelawong and Biswas 2008). However, Betty's Brain uses different content (maps of causal influences) and a different task (teaching an agent with the map). Further research is needed to disentangle if Guru's concept map inconsistency is due to the concept map task or due to the type of knowledge in the maps (e.g., links being remembered most saliently; Cade et al. 2014).

## AutoTutor Technologies: Semantic Analysis, Authoring, and Delivery

AutoTutor research has also developed advanced capabilities for semantic analysis, authoring, and tutoring delivery platforms. Semantic analysis research has been driven by tutoring systems (AutoTutor, iSTART, DeepTutor) and by semantic analysis tools (QUAID, Coh-Metrix, SEMILAR). AutoTutor's authoring tools (ASAT) are also discussed, including a discussion of emerging authoring tool designs. Finally, delivery platforms are considered, such as web-based tutors (AutoTutor Lite, BCRA-Gist), service-oriented architectures (AutoTutor Web Services, Sharable Knowledge Objects), and open-source releases (GnuTutor).

### Semantic Analysis and Natural Language Processing

Semantic analysis is a central technology behind any natural language tutoring system. Early dialog-based tutors were extremely constrained in their ability to understand natural language. The SCHOLAR tutor, noted previously, had very limited language understanding and could get very little out of wrong or off-topic answers (Carbonell 1970). CIRCSIM was one of the first successful natural language tutoring systems and was designed to teach blood pressure regulation to complement a simulation-based environment for cardiology experimentation (Kim et al. 1989). CIRCSIM relied on a domain-specific and hand-made lexicon that mapped to statements about ontology objects, such as "Heart rate is increasing" (Glass 1997). This system did not have much flexibility for transitioning to new domains, primarily due to the approaches used to handle natural language. SOPHIE, a tutoring system for circuit design, used a more powerful technique based on semantic grammars, but developing semantic grammars remains a challenge (Burton 1977; Gavaldà and Waibel 1998).

In many ways, AutoTutor represents a synthesis of studies on naturalistic human tutoring with technologies in computational linguistics, such as latent semantic analysis (Graesser et al. 2007b) and Coh-Metrix (Graesser et al. 2004b). AutoTutor relies on lexicons, part-of-speech classifiers, speech act classifiers, syntactic parsers, regular expressions, templates, corpora, LSA, and other semantic analysis tools (Graesser et al. 2012a, b). This multi-faceted approach is shared by other contemporary natural language ITS families, which often have one central natural language understanding technology supported by complementary methods. For example, the BEETLE II tutor for circuit design combines semantic grammars with lexical databases (e.g., WordNet), reference resolution algorithms, and statistical classifications (Dzikovska et al. 2013).

For AutoTutor, LSA has played a central role because it offers a robust semantic matching algorithm that compares the student's verbal input to AutoTutor's expectations and anticipated misconceptions. The most common semantic matching algorithm currently used by AutoTutor combines LSA, regular expressions, and frequency weighted content word overlap (Cai et al. 2011). Syntactic parsing does not help much in the pattern matching operations because much of the students' verbal input is ungrammatical. However, statistical syntactic parsers are useful for classifying the student contributions into speech act categories. Clearly, AutoTutor must respond differently to speech acts that are assertions versus questions, expressive evaluations, acknowledgements ("okay"), metacognitive expressions ("I'm lost," "I don't know").

LSA supports natural language processing by using a statistical technique called singular-value decomposition to represent each word by a vector of component semantic statistical dimensions (factors). These component factors are generated by processing a large corpus of millions of words, in tens of thousands of documents, with typically 100–500 such dimensions being generated (Landauer et al. 1998). This process uses a bag-of-words model that captures co-occurrence information while ignoring ordering. In essence, LSA captures what words co-occur with other words in naturalistic documents, such as encyclopedias, but not in rigid dictionaries. A domain-specific LSA dimensional model can be created based on word co-occurrence within a specific domain corpus. To compare a student contribution against an anticipated contribution (expectation or misconception), each word is replaced by its dimensional equivalent. All of the dimensions of the words within the contribution are aggregated (typically summed) into a single dimensional vector. The similarity between the actual and anticipated (e.g., ideal) vectors are then calculated, such as by calculating the cosine between the vectors to generate a number ranging from 0 (no match) to 1 (perfect match).

LSA provides a quick technique for evaluating the quality of student contributions, with studies indicating that AutoTutor's evaluations of student contributions performed comparably to intermediate domain experts (Graesser et al. 2000). LSA (and similar approaches) also allow AutoTutor to transition effectively to new domains by training on new corpora and developing semantic pattern matching assessments for new expectations and misconceptions. This does not mean that tutoring a new domain is always accurate, but LSA allows a high degree of automation compared to hand-crafted semantic grammars and parser-based approaches.

The LSA semantic space for the original AutoTutor for computer literacy was generated from two textbooks on the topic, 30 articles, and the curriculum script (Graesser et al. 1999). Before testing the system for instructional purposes, the ability of AutoTutor to evaluate student responses was tested on a sample of 192 answers to questions in the curriculum script. This test revealed a correlation of 0.49 between the LSA evaluations and the average evaluations of four human experts, two with intermediate expertise and two with advanced expertise (Graesser et al. 1999). By comparison, the correlation between the ratings of the intermediate experts (graduate students) was 0.51 and the correlation between the ratings among advanced experts (graduate degree holders active in the topic) was 0.78. As such, the LSA scoring performed comparably with the intermediate experts, who are representative of typical tutors.

Early versions of AutoTutor produced solid learning gains, but evaluating student contributions had room for improvement. Expert ratings of the quality for student statements using AutoTutor for Physics were less robust than those for computer literacy, with expert correlations reported as $r$=0.29 correlation with LSA, $r$=0.25 with Kendall's Tau (word ordering), $r$=0.39 word overlap (words shared between expectation and statement), and $r$=0.42 when all three were combined (Graesser et al. 2007b; Rus and Graesser 2006). LSA has clear limitations due to ignoring sentence syntax and ordering: it cannot handle negations ("correct" vs. "not correct") or resolve terms using context (e.g., "He ran" vs. "Charlie ran"). To overcome these issues, AutoTutor has relied on regular expressions to find word overlap and detect necessary compound structures (Cai et al. 2011, 2012). This yields semantic match scores with a reliability of 0.67 whereas trained humans agree 0.69. More advanced aggregation techniques across words are also possible, such as inverse frequency weighting and contextual weighting (e.g., negations changing the weights for terms). However, this process increases authoring effort and the reliability of the performance is sometimes expectation-specific. Physics hits these problems more than computer literacy, because physics relationships are more abstract (e.g., "x has twice the velocity of y" vs. "The CPU reads instructions from RAM"). These challenges have spurred development of more advanced tools.

Rus et al. (2008) improved on AutoTutor's semantic analysis techniques by detecting textual entailment. Textual entailment of "$X$ entails $Y$" holds true when "If a human read $X$, they would probably infer that $Y$ is true." Entailment uses subsumption relationships (e.g., "Y is a type of X") to determine if the student's statement is a more general version of the expectation. This was executed in real-time by parsing statements into graphs and applying subsumption hierarchically to determine if the entities and relationships from the expectation were present in the student statement (i.e., same terms, synonyms from thesaurus, more general terms). This approach offered an improvement over word-overlap, LSA, and some similar word-to-word entailment measures (Rus et al. 2008, 2009). Detecting entailment and paraphrasing (mutual entailment) was used by iSTART and early versions of DeepTutor (Rus et al. 2009, 2013c).

A number of standalone semantic analysis packages have evolved in tandem with the AutoTutor family. QUAID (Question Understanding Aid) was developed to evaluate the comprehensibility of questions to check if surveys are likely to provide useful results (Graesser et al. 2006). Later, the Coh-Metrix was designed to support over 200 measures of text cohesion, language, and readability (Graesser et al. 2004b; McNamara et al. 2014). Coh-Metrix (www.cohmetrix.com) is a free web-based service intended to analyze passages of text, with a focus on ease versus difficulty of comprehension. Coh-Metrix has been used to analyze AutoTutor conversations (Graesser et al. 2007a) and emotions during tutoring (D'Mello and Graesser 2012a). A recent development is the SEMILAR (SEMantic SimILARity) toolkit, which implements a variety of word-to-word similarity measures, classifiers, and entailment algorithms related to the DeepTutor project (Rus et al. 2013b). A notable algorithm in SEMILAR is DeepTutor's Quadratic Assignment for optimal word-to-word matching, which outperformed all other reported algorithms (77.6 % accuracy; Rus et al. 2013c) on text-to-text similarity measures for sentences in the Microsoft Research Paraphrase corpus (Dolan et al. 2004). The algorithms in SEMILAR are described in Rus et al. (2013b).

## Authoring Tools

The primary authoring tool for AutoTutor is ASAT (AutoTutor Script Authoring Tool), also called ASAT-D (Desktop), whose initial design was outlined by Susarla et al. (2003) but has been substantially revised to the point of being licensed for numerous applications. The new version of this tool streamlines authoring and supports better integration of media (e.g., graphics) and testing of AutoTutor's scripts and conversational rules. Authoring tools are crucial for scaling up intelligent tutoring systems and other advanced learning environments. Domain knowledge is essential for tutoring but domain experts are unlikely to have the technical skills or time to learn complicated interfaces or programming functionality. Time costs of authoring for ITS can be particularly high, with some systems estimating approximately 100 or more hours of authoring time for a single hour of instruction (Koedinger et al. 2004). With ASAT, users with limited technical expertise can author a tutoring script in under 1 h (Song et al. 2004). This is made possible because, unlike tutors that rely on programming or special data representations, authoring AutoTutor scripts primarily involves writing (in natural language) questions, expectations, misconceptions, hints, prompts, summaries, and other verbal content. Nevertheless, technical expertise is needed to handle regular expressions, links to external media requiring adaptive interaction, rule sets for conversations with adaptive complexity, and complex branching.

Significant room for improvement remains for improving three functions of authoring tools: collaborative authoring, reducing the learning curve, and simplifying dialog rule authoring. First, while scripts ("tutoring packs") are modular, ASAT-D has no way for multiple authors to collaboratively author different parts of the same script. Ongoing work with the Sharable Knowledge Objects (SKO) project, described next, is exploring the feasibility of web-based collaborative authoring with ASAT-W (Web-Based), which uses cloud hosting and basic version control (e.g., similar to a Google document). Second, while ASAT-D no explicit programming, learning to use the tool effectively still involves a learning curve related to implicit understanding of programming concepts (e.g., conditions and rules). For in-service teachers and curriculum designers, time-costs can be nontrivial, so a form-based tool called ASAT-FB (Form-Based) is under development to provide a wizard design to streamline this process. Finally, to simplify rule authoring, development of a visual flow-chart authoring interface is under development as ASAT-V (Visual).

## Delivery Technologies and Architectures

Delivery technologies are also very important for AutoTutor and ITS in general. While intelligent tutoring systems are effective teaching tools (Graesser et al. 2012a; VanLehn 2011), wide scale adoption remains a serious challenge for tutoring systems. AutoTutor has steadily worked toward minimizing the technical requirements for end-users of the system, with initial designs relying on installed software, later designs moving to web-based Java applications, and more recent designs using a mixture of HTML and Flash.

AutoTutor Lite implements a constrained version of AutoTutor that is designed for easier authoring (ASAT-W), rapid web deployment, and integration into third-party systems such as games (www.skoonline.org; Hu et al. 2009). This version of AutoTutor does not employ the full range of semantic analysis methods, but is limited to LSA and keyword-based analysis. AutoTutor Lite does not have all of the dialog flexibility of AutoTutor, but its main dialog style still uses deep reasoning questions complemented by simpler self-reflection questions and vicarious tutoring. AutoTutor Lite uses a simplified student model called Learner's Characteristic Curves (LCC) that is based on the relevance and novelty of student contributions (Morrison et al. 2014). Relevance, as with the original AutoTutor, is the similarity between the student's contribution and the expectations. Novelty captures how different the student's contribution is from their prior contributions toward that topic, i.e., are they covering new ground or rehashing old points? These factors trigger discourse moves such as hints, to determine when to move on to new topics, and to select new questions. These restrictions help AutoTutor scale as a web application and also simplify authoring.

Evaluations of learning gains for systems using AutoTutor Lite are underway, as it is being used for the VCAEST military medical training and BRCA-Gist, a tutor for assessing breast cancer risk. The content for BRCA-Gist was authored by researchers at Miami University, making it one of the first AutoTutor descendants authored outside of Memphis. A recent study using BCRA-Gist reported that the AutoTutor Lite's evaluations of student contributions predicted student outcomes on a posttest (Wolfe et al. 2013). A second study indicated that AutoTutor Lite outperformed a static web-based tutorial for reducing internal inconsistency (e.g., negating prior assumptions) but did not improve quantitative estimates of risk probabilities (Wolfe et al. 2012).

Three additional projects have also attempted to make AutoTutor more accessible. The first such project, GnuTutor, was a Java open-source release with elements of AutoTutor and AutoTutor Lite (Olney 2009). GnuTutor was intended for third-party developers who wish to inspect or build on the system. AutoTutor and AutoTutor Lite development has continued since this time so an updated release of GnuTutor is needed in the future. Second, recent versions of AutoTutor have been exposed as the AutoTutor Web Services (ATWS), to improve interoperability with other systems. For example, the University of Wisconsin's *Land Science* multiparty game helps groups of middle school students learn about urban science in their community (Shaffer 2006; Shaffer and Graesser 2010). The group learning environment has an automated AutoMentor with some of the components of AutoTutor, with the tutoring agent's behaviors modeled after the behaviors of human mentors for students using the system (Shaffer and Graesser 2010). The goal is for students to think and act like STEM professionals, but they can only do this with the guidance of either a human mentor (Shaffer 2006) or an AutoMentor. AutoMentor gives each student group suggestions and responds to some of their questions by either answering the questions directly or asking other students to respond to the questions, then assessing the semantic relevance of the replies. Web services help AutoMentor integrate into this multiplayer serious game.

Third, the Sharable Knowledge Objects (SKO) Module system is an ongoing project that can extend AutoTutor or AutoTutor Lite. SKO modules are designed using web-based and service-oriented patterns, allowing each SKO to be an encapsulated tutoring unit that composes local and remote services (Nye 2013). Each SKO is capable of conversation using one or more high-quality agent avatars, text, graphics, movies, sound files, and embedded web resources. The SKO project moves AutoTutor toward a service-oriented design that increases the modularity of tutor designs. This should improve integration into learning management systems and other service-oriented systems, such as the Generalized Intelligent Framework for Tutoring architecture (GIFT; Sottilare et al. 2012). As part of this system, a persistent student model service is being developed to track and report student knowledge levels.

The SKO framework is currently being used to approach Algebra I mathematics, which is a new domain for AutoTutor. This system is being integrated to provide just-in-time tutoring for ALEKS (Assessment and Learning in Knowledge Spaces), an adaptive learning system based on knowledge space theory (Falmagne et al. 2006). Each relevant ALEKS problem will be supported by AutoTutor trialogs (three-way conversations between the user, the tutor, and a simulated student). Algebra is more procedural than many topics approached by AutoTutor in the past, making this a qualitatively different domain. A major goal of the SKO project is to develop a tutoring architecture that can be integrated with existing content (e.g., adding tutoring to existing HTML pages) and platforms (e.g., ALEKS). SKO focuses on integrating with pre-existing content because the cost to develop new content is a common bottleneck for ITS. While this system is currently being used to produce standalone modules for delivery inside other systems, the framework supports real-time semantic messaging to integrate into dynamic systems (e.g., simulations). By making it easier to enhance existing resources with tutoring, it should be possible to make tutoring systems available to a greater number of learners.

## Discussion: Key Findings from AutoTutor

Overall, AutoTutor and related systems have shown learning gains over non-interactive learning materials on a variety of math and science domains: computer literacy, physics, biology, and critical thinking. Average learning gains were approximately 0.8σ (Graesser et al. 2012b, 2008), typically with higher gains for deep learning than shallow. On the one hand, these show an impressive effectiveness as an instructional tool. On the other hand, learning gains fall significantly short of Bloom's (1986) report of 2σ for expert human tutoring and no study reported learning gains in excess of 1.5σ over controls. These results can be interpreted in two ways: either it is extremely hard to match the performance of human tutoring or human tutors are not actually as good as Bloom estimated. Meta-analyses by Cohen et al. (1982) and VanLehn (2011) reported effect sizes of only 0.4σ and 0.79σ for human tutoring, respectively. Studies with gains over 0.8σ tended to be short (<4 weeks) and structured interventions. Learning gains of 2σ might only be possible for special subsets of domain content or extremely effective tutors. With that in mind, AutoTutor is probably more effective than many non-expert tutors and might even be on-par with a typical expert tutor.

The particular mechanisms of such learning gains in AutoTutor-style tutoring have been examined in some studies, but additional work is needed to disentangle the components that are responsible for learning. Studies have examined the effect of disabling or altering specific tutor functions, but it is infeasible to empirically test every combination of tutoring features. Table 4 shows the relative difference in learning

**Table 4** Changes in learning gains under different conditions

| Description | Δ Learning | Study |
|---|---|---|
| *AutoTutor Base System* | 0.80σ | (Average of many) |
| Text only (No avatar + no voice synthesizer) | −0.13σ[1] | Graesser et al. (2004a); Graesser et al. (2008) |
| Voice only (No avatar) | −0.13σ[1] | |
| AutoTutor with Redundant Text + Voice | +0.34σ[1] | |
| Human spoken input (instead of typed text) | ~ | D'Mello et al. (2011a, b) |
| Human reads relevant book sections that answers a question(non-interactive) | −0.22σ | Graesser et al. (2004a) |
| Human reads tutoring scripts directly relevant to the main question (non-interactive) | −0.07σ | |
| Tutoring by an expert tutor over text chat | −0.08σ | |
| Vicarious learning (watch tutoring only) | ~ | Gholson et al. (2009) |
| 3D Simulations (Physics) | +0.22σ[1] | Graesser et al. (2005a, b) |

[1] Not statistically significant

gains when certain features were disabled or altered, with the relevant study results cited. A tilde mark ("~") means that the study found this design reported no statistically significant difference in learning gains from the base AutoTutor system, nor any consistent effect direction. The base system for AutoTutor uses text input from the human and a talking head avatar that uses synthesized speech. This table demonstrates, in part, the difficulty in obtaining statistically significant effects when changing individual features, even when using reasonable sample sizes (25–40 subjects per condition).

The only consistent and reliable differences are seen when comparing AutoTutor against different types of content, such as reading static text or reading the tutoring scripts. These findings indicate that engaging with appropriate, relevant content dominates other factors such as modality and sometimes even interactivity. However, the issue of information equivalence complicates these evaluations: textbooks seldom include deep reasoning questions, so curriculum designers must generate them for AutoTutor. While the similarity between control text and tutoring scripts is high (e.g., LSA cosine similarity=0.58; VanLehn et al. 2007), they are not identical. This issue is systemic: textbooks do not pose deep reasoning questions because they cannot provide feedback and because their rhetorical structure is not organized around deep questions and answers. This may offer a "format advantage" for dynamic media over textbooks, because ITS can present content (e.g., questions) that would not make sense in a textbook.

The small difference (−0.07σ) between AutoTutor versus reading the tutoring scripts belies the importance of adapting to the student. A study of WHY2/Atlas and WHY2/AutoTutor (VanLehn et al. 2007) controlled for content by comparing the human tutors, tutoring systems, canned text (static tutoring script content), and textbooks. They found that human tutors strongly outperformed canned scripts (1.64σ) when novices encountered intermediate content (e.g., content in their zone of proximal development). Ceiling and floor effects were highly evident outside of the zone of proximal development. When students had very high knowledge, tutors (human or computer) performed no better than canned statements. Likewise, when students totally lacked the requisite background knowledge, they performed poorly regardless of the condition (VanLehn et al. 2007). This demonstrates how prior knowledge can confound effect sizes as a measure of effectiveness and might also hinder testing of feature-effectiveness.

Prior achievement can also impact the effectiveness of features directly. Table 5 notes specific dialog strategies and pedagogical agent roles that have significant evidence of being effective (+) versus not significantly effective (~) for learning. In this table, the first six rows (AutoTutor to ARIES) represent unique systems compared against traditional controls as discussed earlier in the paper (e.g., reading texts). The final four rows (AutoTutor-AS to AutoTutor-3D) are modified versions of either AutoTutor or Guru, so they are compared against their respective base system. These findings hint at opportunities to detect student traits that allow tutor features to be more effective, thereby improving personalization of learning. Work with AutoTutor-AS revealed that only low knowledge students benefited from affect sensitivity (D'Mello and Graesser 2012a; D'Mello et al. 2011b). This work mirrors findings with the Wayang Output tutoring system, which found that low knowledge students benefit more from affective support provided by animated agents (Woolf et al. 2010). Similar results have been found for factors such as politeness of animated agents (Wang et al. 2008).

Vicarious dialogs (a tutor agent talking with a peer agent) have also been more effective for low knowledge learners (Craig et al. 2012). While these dialogs can still benefit high knowledge students, they show particular benefits for helping low knowledge students build initial models for domain knowledge. In this way, a vicarious dialog in natural language may be analogous to a worked example for problem-solving. On the converse, interactive simulations were used more productively by higher-performing learners, though increased scaffolding might help other students benefit also (Jackson et al. 2006). While

**Table 5** Effectiveness of tutoring system features on different learners

| Feature | First System w/ Feature | Learner Performance | |
|---|---|---|---|
| | | High | Low |
| Expectation-misconception tutoring dialog (5-frame) | AutoTutor | + | |
| Deep reasoning questions | AutoTutor | + | |
| Collaborative lectures | Guru | + | |
| Navigational agent (search help) | HURAA | ~ | |
| Vicarious agent dialogs | iDRIVE | + | ++ |
| Trialogs (tutor + peer agent) | ARIES | + | |
| Affective sensitivity | AutoTutor-AS (Supportive) | ~ | + |
| Informal interaction | AutoTutor-AS (Shakeup) | ~ | |
| Reacting to disengaged eye contact | Gaze Tutor (Built on Guru) | + | |
| Interactive simulations | AutoTutor-3D | + | ~ |

it is not noted in the table because conclusive evidence is not yet available, it is hypothesized that teachable agents will be most beneficial for high-knowledge learners. This is because teachable agents require the highest levels of learner contributions, since the human student leads the explanation.

Other strategies and features appear effective (or ineffective) for a variety of learners. Expectation-misconception tailored dialogs and deep reasoning questions have been effective for a variety of systems and domains (Graesser et al. 2012b). Guru demonstrated that collaborative lectures are consistently effective (Olney et al. 2012). Reacting to disengagement in GazeTutor was also found to improve deep learning (D'Mello, Olney et al. 2012). Pedagogical agents have also failed in particular roles. For example, the HURAA and SEEK systems indicated that intelligent agents may not add much value as navigators in a hypermedia system (Hu and Graesser 2004). Similarly, the buddy-style AutoTutor-AS Shakeup tutor was less effective than the more traditional Supportive AutoTutor (D'Mello, Lehman, & Graesser 2011). Students' liking of an AutoTutor agent also had no correlation with learning as reported by Moreno et al. (2002), whereas Jackson and Graesser (2007) reported a negative relationship between conditions that promote liking versus deep learning. Studies on vicarious learning have identified some conditions when the human does not even need to be directly interacting with the tutor, but can learn fairly well by watching a simulated student learn (Gholson et al. 2009; Chi et al. 2008). In total, this suggests that pedagogical agents do not need to be navigators or buddies to the learner, nor even liked by the learner. It also indicates that they do not always need to be tutors to the student, but can teach indirectly instead.

Other work suggests that while optimizing microsteps can improve learning gains (Chi et al. 2014), ITS may over-emphasize high levels of interactivity at the micro-step level (VanLehn 2011). Simply stated, the best dialog is not always the most interactive one. This point is supported by Kopp et al.'s (2012) finding on the dosage of interactive tutoring. They reported significant learning gains (0.46σ) for mixing intense dialog on some problems and no dialog in others when compared to a system that used intense dialog exclusively. Trialog-based environments like Operation ARIES/ARA allow great flexibility in tutoring exchanges to explore the tradeoffs of mixing direct tutoring, vicarious tutoring, teachable agents, and traditional media (Forsyth et al. 2012; Halpern et al. 2012; Millis et al. 2011).

It is possible that the optimal level of interactivity depends on the students' knowledge level. Table 5 hints at a trend where low-knowledge students benefit most from low control over the interaction (e.g., vicarious) and high-knowledge students benefit from high control over the interaction (e.g., simulations). If this trend holds true, we would expect a positive correlation between knowledge level and the benefits from interactivity, such as: Vicarious (lowest knowledge) → Collaborative Lectures → Deep Questions and Expectation-Misconception Dialog → Teachable Agents → Simulations (highest knowledge). While this pattern intuitively makes sense, many of these strategies have not been directly compared, so further research is needed to see if this holds empirically. Moving forward, new roles and arrangements for ensembles of pedagogical agents should continue to enhance the quality of tutoring systems by allowing researchers to analyze and compare different tutoring strategies.

## Future Directions

This review of AutoTutor's history demonstrates that conversational tutoring systems can help students with a variety of domains and subject matter. In many ways, the success of conversational tutors across many subjects validates a core idea behind AutoTutor: that the conversational agent offers a universal interface for tutoring that can implement a variety of pedagogical strategies. Assuming that natural language tutoring can be universal, can it become ubiquitous? The evolution of AutoTutor shows a gradual shift in this direction. The initial AutoTutor design was a standalone desktop application, but is becoming available as standards-based web services. The AutoTutor family has integrated increasingly rich and diverse media into their tutoring environments, including simulations, interactive concept maps, and videos. In some recent projects, the roles of AutoTutor agents and complementary media have reversed: tutoring agents are being embedded into static media (e.g., HTML pages) and dynamic media (e.g., virtual worlds). Considering conversational agents as a universal interface, this inversion offers powerful opportunities for embedding tutoring agents into the wealth of existing digital educational media and learning systems. Conversational interfaces are also uniquely suited for mobile learning. For example, natural language tutors could be effective as situated tutors, such as mobile learning tutors that are triggered by GPS hotspots (Hwang and Tsai 2011).

Despite clearing technological hurdles, transferring tutoring systems into schools and commercial use has proved difficult for the ITS field in general. Learning gains are only one of many considerations for teachers and administrators, making it harder to get ITS into the hands of students. While thousands of learners have used AutoTutor, there has been little sustained use by K-12 schools. However, AutoTutor and related systems have recently made promising footholds. Projects such as Guru and DeepTutor were each recently evaluated by over 200 students in K-12 classrooms, and have potential for broader use. The CSAL project for adult literacy will reach 500 adult learners during its evaluation and could make a large impact as a web service, since nearly 3 million Americans enroll in adult literacy programs annually. Partnerships with commercial groups are also growing. A specialized version of ASAT for Assessment (ASATA) is being developed with the Educational Testing Service (ETS). Operation ARA is being expanded with Pearson for use as a serious game. Finally, SKO modules for Algebra are being integrated with ALEKS, a commercial learning system with McGraw-Hill that has served millions of students over more than a decade. AutoTutor is also integrated into the Generalized Framework for Intelligent Tutoring (GIFT) architecture, a major Army Research Lab project working to build the next generation of tutoring standards and authoring (Sottilare et al. 2012).

Getting tutoring systems into the hands of learners is particularly important because there is little doubt that tutoring systems are effective. Studies indicate that tutoring systems may have surpassed non-expert tutors and might even match expert human tutors on some topics (Graesser et al. 2012a; VanLehn 2011). This leads to an important question about the future of tutoring systems: what if human tutors were not as effective as we previously thought? Should tutoring systems attempt to model more effective human tutors (e.g., Guru Tutor)? Alternatively, should ITS attempt to take advantage of optimized micro-steps or idealized progressions that human tutors would have difficulty implementing? Finally, how tutoring systems be optimized to promote greater use by students and classes (e.g., SKO Modules)? There are good reasons to pursue each of these research agendas, as well as significant questions about which types of students benefit most from certain tutoring features. As research based on AutoTutor continues, new projects will continue to explore the principles that produce effective and accessible learning.

**Acknowledgments** This work has been supported by grants from the National Science Foundation (NSF grants 0325428, 633918, 0834847, 0918409, 1108845, 0834847, 0904909, 0918409, 1108845), Institute of Education Sciences (IES grants R305A080594, R305G020018, R305C120001, R305A090528, R305A100875), Office of Naval Research (ONR STEM Grand Challenge: N00014-12-C-0643), Army Research Lab (N00014-12-C-0643), and the Educational Testing Service. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of these funding sources.

## Appendix 1: Glossary of Projects

**ASAT:** AutoTutor Script Authoring Tool is the primary authoring tool for AutoTutor. Can direct multiple agents and external events/controls.

**ASATA:** AutoTutor Script Authoring Tool for Assessment is a specialized authoring tool developed with the Educational Testing Service for developing for building dialog-based high stakes assessments.

**AutoMentor *(STEM Thinking)*:** Uses epistemic analysis of discourse in student group chats to help students learn how to think and act like STEM (science, technology, engineering, and mathematics) professionals in a multi-party serious game simulation of urban planning.

**AutoTutor *(Computer Literacy)*:** Core AutoTutor natural language tutoring system, which uses expectation-misconception dialog and deep questions, latent semantic analysis & regular expressions, and talks with user through the animated agent(s).

**AutoTutor-3D *(Physics)*:** An extension of AutoTutor for physics, AutoTutor-3D added interactive three dimensional simulations of physics problems designed in 3D Studio Max.

**AutoTutor Affect-Sensitive *(Computer Literacy)*:** AutoTutor-AS detected affect using natural language and discourse, facial expressions, body posture, and speech. Feedback considered student emotions and cognitive states. Sometimes called AutoTutor-ES (Emotion Sensitive).

**AutoTutor Lite *(General)*:** AutoTutor Lite (ATL) is a web-based variant of AutoTutor designed for simpler authoring, rapid deployment, and integration into third-party systems.

**BRCA-Gist *(Breast Cancer Risk)*:** An AutoTutor Lite tutor led by the Miami University, intended to tutor understanding of risk probabilities and personal breast cancer risk.

**Coh-Metrix:** A linguistic analysis toolkit with over 200 metrics. The "Coh" stands for cohesion and coherence.

**CSAL Adult Literacy Tutor *(Reading)*:** This tutoring system project for the Center for the Study of Adult Literacy (CSAL) is intended to help learners who struggle with print media, through closer integration of trialogs, web pages, and multimedia.

**DeepTutor** ***(Physics)*****:** Tutor that uses learning progressions to foster deep learning of physics concepts, as well as enhanced semantic analysis, such as entailment.

**GazeTutor** ***(Biology)*****:** Enhanced version of Guru Tutor that monitors and reacts to student gaze.

**Gnu Tutor** ***(General)*****:** An open source Java release of an early version AutoTutor Lite.

**Guru Tutor** ***(Biology)*****:** Tutoring system for biology designed based on observation of expert tutors. Uses collaborative lecturing and concept maps to support learning.

**HURAA** ***(Research Ethics)***: The Human Use Regulatory Affairs Advisor for training ethics in human experiments. AutoTutor agents helped navigate hypertext multimedia containing case-based reasoning and multiple information retrieval mechanisms.

**iDRIVE** ***(Computer Literacy, Physics, Biology)*****:** Instruction with Deep-Level Reasoning Questions in Vicarious Environments where the learner observes two pedagogical agents demonstrate deep explanations and model effective learning behavior (e.g. question-asking).

**iSTART** ***(Reading)*****:** Interactive Strategy Training for Active Reading and Thinking is a tutoring system for improving reading comprehension by training reading strategies. Uses multi-agent conversations and specialized semantic analysis to tutor reading strategies.

**iSTART-ME** ***(Reading)*****:** The Motivationally-Enhanced (ME) version of iSTART provides tutoring using an interactive game environment.

**MetaTutor** ***(Biology)*****:** Tutors self-regulated learning (SRL) skills inside a hypermedia setting.

**Operation ARA** ***(Scientific Reasoning)*****:** Operation Acquiring Research Acumen is an extension of the Operation ARIES project that adds additional features and game content.

**Operation ARIES** ***(Scientific Reasoning)*****:** Operation Acquiring Research, Investigative, and Evaluative Skills is a trialog-based tutoring system and serious game for teaching critical thinking. Learners resolve inconsistent information about scientific methods inside a serious game narrative.

**QUAID:** Question Understanding Aid was a tool to evaluate the comprehensibility of questions.

**SEEK Web Tutor** ***(Critical Thinking)*****:** The Source, Evidence, Explanation, and Knowledge Tutor was designed to help learners evaluate the credibility and relevance of information using tutoring-enhanced web search, with spoken hints, pop-up ratings and metacognitive journaling.

**SKO Modules** ***(General)*****:** Sharable Knowledge Object Modules are encapsulated, cloud-hosted modules that compose web services to provide tutoring. Currently being applied to Algebra.

**VCAEST** ***(Medical)*****:** Virtual Civilian Aeromedical Evacuation Sustainment Training is designed to train civilian medical personnel on federal guidelines for emergency situations and triage.

**WHY2/AutoTutor** ***(Physics)***: Extension of AutoTutor that approached tutoring conceptual physics. This was part of a larger WHY2 project that included

WHY2/Atlas. WHY2 was a reference to an old tutoring system called WHY and the year 2000 (e.g., Y2K).

**Writing-Pal** ***(Writing)*****:** This tutor attempts to improve essay and academic writing skills and provides automated evaluation and feedback on essays. It is related to the iSTART system.

## Appendix 2

**Table 6** Main tutoring discourse styles, by system

| System[a] | Deep Question | Vicarious | Trialog Styles[b] | Collaborative Lectures | Navigation & Queries | Other |
|---|---|---|---|---|---|---|
| ARA/ARIES | X | X | X | | | |
| AutoTutor | X | | | | | |
| AutoTutor-3D | X | | | | | X[c] |
| AutoTutor-AS | X | | | | | |
| AutoTutor Lite | X | X | | | | |
| BRCA-Gist | X | X | | | | |
| CSAL | X | X | X | | | X[d] |
| DeepTutor | X | | | | | |
| GazeTutor | | | | X | | |
| GnuTutor | X | | | | | |
| GuruTutor | | | | X | | |
| HURAA | X | | | | X | |
| iDRIVE | (X)[e] | X | | | | |
| MetaTutor | X | | | | X | X[f] |
| SEEK | X | | | | X | |
| SKO | X | X | X | | | |
| VCAEST | X | | | | | X |
| WHY2/AutoTutor | X | | | | | |

[a] iSTART, iSTART-ME, and Writing-Pal are not listed because their tutoring styles differ significantly from other tutors, due to their focus on tutoring reading and writing skills

[b] Multiple trialog styles exist, including teachable agents, trialog-based deep reasoning questions, and peer competition quiz games such as those used by Operation ARIES. Not all trialog-capable systems use all available modes in practice, but all are able to use them

[c] AutoTutor-3D asked for predictions about simulations and responded to those predictions

[d] CSAL uses click-based inputs, such as interactive graphics, to help improve literacy

[e] iDRIVE presented deep questions asking and answering vicariously

[f] MetaTutor had multiple agents that served distinct roles for fostering metacognition

## References

Aleven, V., Mclaren, B. M., Sewall, J., & Koedinger, K. R. (2009). A new paradigm for intelligent tutoring systems: example-tracing tutors. *International Journal of Artificial Intelligence in Education, 19*(2), 105–154.

Anderson, L. W., & Krathwohl, D. R. (2001). *A taxonomy for learning, teaching, and assessing*. New York: Longman.

Azevedo, R., Johnson, A., Chauncey, A., & Burkett, C. (2010). Self-regulated learning with MetaTutor: Advancing the science of learning with MetaCognitive tools. In M. S. Khine & I. M. Saleh (Eds.), *New science of learning* (pp. 225–247). New York: Springer.

Azevedo, R., Landis, R. S., Feyzi-Behnagh, R., Duffy, M., Trevors, G., Harley, J. M., & Hossain, G. (2012). The effectiveness of pedagogical agents' prompting and feedback in facilitating co-adapted learning with MetaTutor. In S. A. Cerri & B. Clancey (Eds.), *Proceedings of Intelligent Tutoring Systems (ITS) 2012* (pp. 212–221). Berlin: Springer.

Bloom, B. S. (1956). *Taxonomy of educational objectives, handbook I: The cognitive domain*. New York: David McKay Co, Inc.

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*(6), 4–16.

Brophy, S., Biswas, G., Katzlberger, T., Bransford, J., & Schwartz, D. (1999). Teachable agents: Combining insights from learning theory and computer science. In S. P. Lajoie & M. Vivet (Eds.), *Proceedings of Artificial Intelligence in Education (AIED) 1999* (pp. 21–28). Amsterdam: IOS Press.

Burton, R. R. (1977). Semantic grammar: an engineering technique for constructing natural language understanding systems. *ACM SIGART Bulletin, 61*, 26.

Cade, W. L., Maass, J. K., Hays, P., & Olney, A. M. (2014). Animated presentation of pictorial and concept map media in biology. In *Intelligent tutoring systems* (pp. 416–425). Berlin: Springer.

Cai, Z., Forsyth, C., Germany, M. L., Graesser, A. C., & Millis, K. (2012). Accuracy of tracking student's natural language in OperationARIES!: A serious game for scientific methods. In S. A. Cerri & B. Clancey (Eds.), *Proceedings of Intelligent Tutoring Systems (ITS) 2012* (pp. 629–630). Berlin: Springer.

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

Carbonell, J. R. (1970). AI in CAI: an artificial-intelligence approach to computer-assisted instruction. *Man–machine Systems, IEEE Transactions on, 11*(4), 190–202.

Chi, M. T., De Leeuw, N., Chiu, M. H., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science, 18*(3), 439–477.

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

Chi, M., Jordan, P., & VanLehn, K. (2014). When is tutorial dialogue more effective than step-based tutoring? In S. Trausan-Matu, K. E. Boyer, M. Crosby, & K. Panourgia (Eds.), *Proceedings of Intelligent Tutoring Systems (ITS) 2014*. Berlin: Springer.

Cohen, P. A., Kulik, J. A., & Kulik, C. L. C. (1982). Educational outcomes of tutoring: a meta-analysis of findings. *American Educational Research Journal, 19*(2), 237–248.

Craig, S. D., Gholson, B., Brittingham, J. K., Williams, J. L., & Shubeck, K. T. (2012). Promoting vicarious learning of physics using deep questions with explanations. *Computers & Education, 58*(4), 1042–1048.

Craig, S. D., Sullins, J., Witherspoon, A., & Gholson, B. (2006). The deep-level-reasoning-question effect: the role of dialogue and deep-level-reasoning questions during vicarious learning. *Cognition and Instruction, 24*(4), 565–591.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science, 41*(6), 391–407.

Dehn, D. M., & Van Mulken, S. (2000). The impact of animated interface agents: a review of empirical research. *International Journal of Human-Computer Studies, 52*(1), 1–22.

Dillenbourg, P., & Traum, D. (2006). Sharing solutions: persistence and grounding in multi-modal collaborative problem solving. *The Journal of the Learning Sciences, 15*, 121–151.

D'Mello, S., Picard, R., & Graesser, A. (2007). Towards an affect-sensitive autotutor. *IEEE Intelligent Systems, 22*(4), 53–61.

D'Mello, S. K., Dowell, N., & Graesser, A. C. (2011a). Does it really matter whether students' contributions are spoken versus typed in an intelligent tutoring system with natural language? *Journal of Experimental Psychology: Applied, 17*(1), 1–17.

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

D'Mello, S. K., & Graesser, A. C. (2012a). AutoTutor and affective AutoTutor: Learning by talking with cognitively and emotionally intelligent computers that talk back. *ACM Transactions on Interactive Intelligent Systems*, *2*(4), 23:2–23:29.

D'Mello, S. K., & Graesser, A. C. (2012). Dynamics of affective states during complex learning. *Learning and Instruction, 22*(2), 145–157.

D'Mello, S. K., Hays, P., Williams, C., Cade, W., Brown, J., & Olney, A. (2010a). Collaborative lecturing by human and computer tutors. In V. Aleven, J. Kay, & J. Mostow (Eds.), *Proceedings of Intelligent Tutoring Systems (ITS) 2010* (pp. 178–187). Berlin: Springer.

D'Mello, S. K., Lehman, B., & Graesser, A. (2011b). A motivationally supportive affect-sensitive AutoTutor. In R. Calvo & S. D'Mello (Eds.), *New perspectives on affect and learning technologies* (pp. 113–126). New York: Springer.

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

D'Mello, S. K., Olney, A., & Person, N. (2010b). Mining collaborative patterns in tutorial dialogues. *Journal of Educational Data Mining, 2*(1), 1–37.

D'Mello, S., Olney, A., Williams, C., & Hays, P. (2012b). Gaze tutor: a gaze-reactive intelligent tutoring system. *International Journal of Human-Computer Studies, 70*(5), 377–398.

Dolan, B., Quirk, C., & Brockett, C. (2004). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th International Conference on Computational Linguistics* (p. 350–357). Stroudsburg, PA: Association for Computational Linguistics.

Dzikovska, M. O., Farrow, E., & Moore, J. D. (2013). In H. C. Lane, K. Yacef, J. Mostow, & P. Pavlik (Eds.), *Proceedings of Artificial Intelligence in Education (AIED) 2013* (pp. 279–288). Berlin: Springer.

Epstein, R., Roberts, G., & Beber, G. (2009). *Parsing the Turing test: Philosophical and methodological issues in the quest for the thinking computer*. Berlin: Springer.

Falmagne, J. C., Cosyn, E., Doignon, J. P., & Thiéry, N. (2006). The assessment of knowledge, in theory and in practice. In R. Missaoui & J. Schmidt (Eds.), *Formal concept analysis* (pp. 61–79). Berlin: Springer.

Freedman, R., Haggin, N., Nacheva, D., Leahy, T., & Stilson, R. (2004). Using a domain-independent reactive planner to implement a medical dialogue system. In T. Bickmore (Ed.), *AAAI fall symposium on systems for health communication* (pp. 24–31). Menlo Park: AAAI Press.

Forsyth, C. M., Pavlik, P., Graesser, A. C., Cai, Z., Germany, M., Millis, K., Butler, H., & Dolan, R. (2012). In K. Yacef, O. Zaïane, H. Hershkovitz, M. Yudelson, & J. Stamper (Eds.), *Proceedings of the 5th International Conference on Educational Data Mining* (pp. 172–175). Chania: International Educational Data Mining Society.

Gholson, B., Witherspoon, A., Morgan, B., Brittingham, J. K., Coles, R., Graesser, A. C., & Craig, S. D. (2009). Exploring the deep-level reasoning questions effect during vicarious learning among eighth to eleventh graders in the domains of computer literacy and Newtonian physics. *Instructional Science, 37*(5), 487–493.

Glass, M. (1997). Some phenomena handled by the CIRCSIM-Tutor Version 3 input understander. In D. Dankel (Ed.), *Proceedings of the Florida Artificial Intelligence Research Symposium (FLAIRS) 1997* (pp. 21–25). Menlo Park: AAAI Press.

Gavaldà, M., & Waibel, A. (1998). Growing semantic grammars. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, 1* (pp. 451–456). New Brunswick: Association for Computational Linguistics.

Graesser, A. C. (2009). Inaugural editorial for *Journal of Educational Psychology. Journal of Educational Psychology, 101*, 259–261.

Graesser, A. C. (2011). Learning, thinking, and emoting with discourse technologies. *The American Psychologist, 66*(8), 743–757.

Graesser, A. C., Cai, Z., Louwerse, M., & Daniel, F. (2006). Question Understanding Aid (QUAID): a web facility that helps survey methodologists improve the comprehensibility of questions. *Public Opinion Quarterly, 70*, 3–22.

Graesser, A. C., Chipman, P., Haynes, B. C., & Olney, A. (2005a). AutoTutor: an intelligent tutoring system with mixed-initiative dialogue. *Education, IEEE Transactions on, 48*(4), 612–618.

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

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

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

Graesser, A. C., Jackson, G. T., Mathews, E. C., Mitchell, H. H., Olney, A., Ventura, M., & Tutoring Research Group. (2003a). Why/AutoTutor: A test of learning gains from a physics tutor with natural language dialog. In R. Alterman & D. Hirsh (Eds.), *Proceedings of the 25rd Annual Conference of the Cognitive Science Society* (pp. 1–5). Boston: Cognitive Science Society.

Graesser, A. C., Jeon, M., & Dufty, D. (2008). Agent technologies designed to facilitate interactive knowledge construction. *Discourse Processes, 45*(4–5), 298–322.

Graesser, A. C., Jeon, M., Yang, Y., & Cai, Z. (2007a). Discourse cohesion in text and tutorial dialogue. *Information Design Journal, 15*, 199–213.

Graesser, A. C., Li, H., & Forsyth, C. (2014). Learning by communicating in natural language with conversational agents. *Current Directions in Psychological Science*. In press.

Graesser, A. C., Lin, D., & D'Mello, S. (2010). Computer learning environments with agents that support deep comprehension and collaborative reasoning. In M. Banich & D. Caccamise (Eds.), *Generalization of Knowledge: Multidisciplinary Perspectives* (pp. 201–223). New York: Psychology Press.

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

Graesser, A. C., McNamara, D. S., Louwerse, M. M., & Cai, Z. (2004b). Coh-Metrix: analysis of text on cohesion and language. *Behavioral Research Methods, Instruments, and Computers, 36*, 193–202.

Graesser, A. C., McNamara, D. S., & VanLehn, K. (2005b). Scaffolding deep comprehension strategies through Point & Query, AutoTutor, and iSTART. *Educational Psychologist, 40*(4), 225–234.

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

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

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

Graesser, A. C., Person, N. K., & Magliano, J. P. (1995). Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology, 9*, 1–28.

Graesser, A. C., VanLehn, K., Rosé, C. P., Jordan, P. W., & Harter, D. (2001b). Intelligent tutoring systems with conversational dialogue. *AI Magazine, 22*(4), 39–51.

Graesser, A.C., Ventura, M., Jackson, G.T., Mueller, J., Hu, X., & Person, N. (2003). The impact of conversational navigational guides on the learning, use, and perceptions of users of a web site. *Proceedings of the AAAI Spring Symposium 2003 on Agent-mediated Knowledge Management*. Menlo Park, CA: AAAI Press.

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

Graesser, A. C., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., & Tutoring Research Group, & Person, N. (2000). Using latent semantic analysis to evaluate the contributions of students in AutoTutor. *Interactive Learning Environments, 8*(2), 129–147.

Graesser, A. C., Wiley, J., Goldman, S. R., O'Reilly, T., Jeon, M., & McDaniel, B. (2007c). SEEK Web tutor: fostering a critical stance while exploring the causes of volcanic eruption. *Metacognition and Learning, 2* (2–3), 89–105.

Halpern, D. F., Millis, K., Graesser, A. C., Butler, H., Forsyth, C., & Cai, Z. (2012). Operation ARA: a computerized learning game that teaches critical thinking and scientific reasoning. *Thinking Skills and Creativity, 7*(2), 93–100.

Hestenes, D., Wells, M., & Swackhamer, G. (1992). Force concept inventory. *The Physics Teacher, 30*, 141–158.

Hu, X., Cai, Z., Han, L., Craig, S. D., Wang, T., & Graesser, A. C. (2009). AutoTutor Lite. In V. Dimitrova, R. Mizoguchi, B. Du Boulay, & A. C. Graesser (Eds.), *Proceedings of Artificial Intelligence in Education (AIED) 2009* (p. 802). Amsterdam: IOS Press.

Hu, X., & Graesser, A. C. (2004). Human use regulatory affairs advisor (HURAA): learning about research ethics with intelligent learning modules. *Behavior Research Methods, Instruments, & Computers, 36*(2), 241–249.

Hwang, G. J., & Tsai, C. C. (2011). Research trends in mobile and ubiquitous learning: a review of publications in selected journals from 2001 to 2010. *British Journal of Educational Technology, 42*(4), E65–E70.

Jackson, G. T., Dempsey, K. B., & McNamara, D. S. (2010). The evolution of an automated reading strategy tutor: From the classroom to a game-enhanced automated system. In M. S. Khine & I. M. Saleh (Eds.), *New science of learning* (pp. 283–306). New York: Springer.

Jackson, G. T., Dempsey, K. B., & McNamara, D. S. (2011). Short and long term benefits of enjoyment and learning within a serious game. In G. Biswas, S. Bull, J. Kay, & A. Mitrovic (Eds.), *Proceedings of Artificial Intelligence in Education (AIED) 2011* (pp. 139–146). Berlin: Springer.

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

Jackson, G. T., & Graesser, A. C. (2007). Content matters: An investigation of feedback categories within an ITS. In R. Luckin, K. Koedinger, & J. Greer (Eds.), *Proceedings of Artificial Intelligence in Education (AIED) 2007* (pp. 127–134). Amsterdam: IOS Press.

Jackson, G. T., Olney, A., Graesser, A. C., & Kim, H. J. (2006). AutoTutor 3-D simulations: Analyzing user's actions and learning trends. In R. Son (Ed.), *Proceedings of the 28th Annual Meeting of the Cognitive Science Society* (pp. 1557–1562). Mahwah: Erlbaum.

Johnson, W. L., & Rickel, J. (1997). Steve: an animated pedagogical agent for procedural training in virtual environments. *ACM SIGART Bulletin, 8*(1–4), 16–21.

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*(1), 47–78.

Jordan, P. W., Makatchev, M., Pappuswamy, U., VanLehn, K., & Albacete, P. L. (2006). A natural language tutorial dialogue system for physics. In G. Sutcliffe & R. Goebel (Eds.), *Proceedings of Florida Artificial Intelligence Research Society Conference (FLAIRS) 2006* (pp. 521–526). Menlo Park: AAAI Press.

Kim, H. J., Graesser, A. C., Jackson, G. T., Olney, A., & Chipman, P. (2005). The effectiveness of computer simulations in a computer-based learning environment. In *Proceedings for e-Learn 2005: World Conference on E-learning in Corporate*, *Government*, *Healthcare*, *and Higher Education* (pp. 1362–1367). Vancouver: AACE.

Kim, N., Evens, M., Michael, J. A., & Rovick, A. A. (1989). CIRCSIM-TUTOR: An intelligent tutoring system for circulatory physiology. In H. Maurer (Ed.), *Proceedings of Computer Assisted Learning, 360* (pp. 254–266). Berlin: Springer.

Klahr, D. (2002). *Exploring science: The cognition and development of discovery processes*. Cambridge: MIT Press.

Koedinger, K. R., Aleven, V., Heffernan, N., McLaren, B., & Hockenberry, M. (2004). Opening the door to non-programmers: Authoring intelligent tutor behavior by demonstration. In J. Lester, R. Vicari, & F. Paraguaçu (Eds.), *Proceedings of Intelligent Tutoring Systems (ITS) 2004* (pp. 162–174). Berlin: Springer.

Koedinger, K. R., Aleven, V., Roll, I., & Baker, R. (2009). In vivo experiments on whether supporting metacognition in intelligent tutoring systems yields robust learning. In D. J. Hacker, J. Dunlosky, & A. C. Graesser (Eds.), *Handbook of metacognition in education* (The Educational Psychology Series, pp. 897–964). New York: Routledge.

Kopp, K. J., Britt, M. A., Millis, K., & Graesser, A. C. (2012). Improving the efficiency of dialogue in tutoring. *Learning and Instruction, 22*(5), 320–330.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes, 25*(2–3), 259–284.

Langston, M. C., & Graesser, A. C. (1993). The "Point and Query" interface: exploring knowledge by asking questions. *Journal of Educational Multimedia and Hypermedia, 2*(4), 355–367.

Leelawong, K., & Biswas, G. (2008). Designing learning by teaching agents: the Betty's Brain system. *International Journal of Artificial Intelligence in Education, 18*(3), 181–208.

Lehman, B., D'Mello, S. K., Strain, A., Mills, C., Gross, M., Dobbins, A., Wallace, P., Millis, K., & Graesser, A. C. (2013). Inducing and tracking confusion with contradictions during complex learning. *International Journal of Artificial Intelligence in Education, 22*, 85–105.

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

Lester, J. C., Towns, S. G., & Fitzgerald, P. J. (1998). Achieving affective impact: visual emotive communication in lifelike pedagogical agents. *International Journal of Artificial Intelligence in Education, 10*, 278–291.

Link, K. E., Kreuz, R. J., & Graesser, A. C. (2001). Factors that influence the perception of feedback delivered by a pedagogical agent. *International Journal of Speech Technology, 4*(2), 145–153.

Long, Y., & Aleven, V. (2013). Skill diaries: Improve student learning in an intelligent tutoring system with periodic self-assessment. In H. C. Lane, K. Yacef, J. Mostow, & P. Pavlik (Eds.), *Proceedings of Artificial Intelligence in Education (AIED) 2013* (pp. 219–228). Berlin: Springer.

Lu, X., Di Eugenio, B., Kershaw, T. C., Ohlsson, S., & Corrigan-Halpern, A. (2007). Expert vs. non-expert tutoring: Dialogue moves, interaction patterns and multi-utterance turns. In *Computational Linguistics and Intelligent Text Processing* (pp. 456–467). Berlin: Springer.

Magliano, J. P., Todaro, S., Millis, K., Wiemer-Hastings, K., Kim, H. J., & McNamara, D. S. (2005). Changes in reading strategies as a function of reading training: a comparison of live and computerized training. *Journal of Educational Computing Research, 32*(2), 185–208.

Massaro, D. W. (1998). *Perceiving talking faces: From speech perception to a behavioral principle* (Vol. 1). Cambridge: MIT Press.

McNamara, D. S., Graesser, A. C., McCarthy, P. M., & Cai, Z. (2014). *Automated evaluation of text and discourse with Coh-Metrix*. Cambridge: Cambridge University Press.

McNamara, D. S., & Magliano, J. P. (2009). Self-explanation and metacognition. In D. Hacker, J. Donlosky, & A. C. Graesser (Eds.), *Handbook of metacognition in education* (pp. 60–81). New York: Taylor & Francis.

McNamara, D. S., O'Reilly, T. P., Best, R. M., & Ozuru, Y. (2006). Improving adolescent students' reading comprehension with iSTART. *Journal of Educational Computing Research, 34*(2), 147–171.

McNamara, D. S., O'Reilly, T., Rowe, M., Boonthum, C., & Levinstein, I. B. (2007). iSTART: A web-based tutor that teaches self-explanation and metacognitive reading strategies. Reading comprehension strategies: Theories, interventions, and technologies, 397–421.

McNamara, D. S., Raine, R., Roscoe, R., Crossley, S., Jackson, G. T., Dai, J., & Graesser, A. C. (2012). The Writing-Pal: Natural language algorithms to support intelligent tutoring on writing strategies. *Applied natural language processing and content analysis: Identification, investigation, and resolution*, (pp. 298–311). Hershey: IGI Global.

Metcalfe, J., & Kornell, N. (2005). A region or proximal of learning model of study time allocation. *Journal of Memory and Language, 52*, 463–477.

Merrill, D. C., Reiser, B. J., Ranney, M., & Trafton, J. G. (1992). Effective tutoring techniques: a comparison of human tutors and intelligent tutoring systems. *The Journal of the Learning Sciences, 2*(3), 277–305.

Millis, K., Forsyth, C., Butler, H., Wallace, P., Graesser, A., & Halpern, D. (2011). Operation ARIES!: A serious game for teaching scientific inquiry. In *Serious games and edutainment applications* (pp. 169–195). London: Springer.

Morrison, D., Nye, B. D., & Hu, X. (2014). Where in the data stream are we? Analyzing the flow of text in dialogue-based systems for learning. In R. Sottilare, A. Graesser, X. Hu, & B. Goldberg (Eds.), *Design recommendations for intelligent tutoring systems: Instructional management* (pp. 237–247). Orlando: U.S. Army Research Laboratory.

Nesbit, J. C., & Adesope, O. O. (2006). Learning with concept and knowledge maps: a meta-analysis. *Review of Educational Research, 76*(3), 413–448.

Nwana, H. S. (1990). Intelligent tutoring systems: an overview. *Artificial Intelligence Review, 4*(4), 251–277.

Nwana, H. S. (1996). Software agents: an overview. *Knowledge Engineering Review, 11*(3), 205–244.

Nye, B. D. (2013). Integrating GIFT and AutoTutor with Sharable Knowledge Objects (SKO). In R. A. Sottilare, & H. K. Holden (Eds.), *Proceedings of the Artificial Intelligence in Education (AIED) 2013 Workshop on the Generalized Intelligent Framework for Tutoring (GIFT)*, (pp. 54–61). CEUR.

Nye, B. D., Graesser, A. C., & Hu, X. (2014a). Multimedia learning with intelligent tutoring systems. In R. Mayer (Ed.), *Multimedia learning* (3rd ed., pp. 703–728). New York: Cambridge University Press.

Nye, B. D., Hajeer, M., Forsyth, C., Samei, B., Hu, X., & Millis, K. (2014). Exploring real-time student models based on natural-language tutoring sessions: A look at the relative importance of predictors. In Z. Pardos and J. Stamper (eds.), *Educational Data Mining (EDM) 2014, (pp.253–256).*

Olney, A. M. (2009). GnuTutor: An open source intelligent tutoring system. In V. Dimitrova, R. Mizoguchi, B. Du Boulay, & A. C. Graesser (Eds.), *Proceedings of Artificial Intelligence in Education (AIED) 2009* (p. 803). Amsterdam: IOS Press.

Olney, A., D'Mello, S., Person, N., Cade, W., Hayes, P., Williams, C., Lehman, B., & Graesser, A. C. (2012). Guru: A computer tutor that models expert human tutors. In S. A. Cerri & B. Clancey (Eds.), *Proceedings of Intelligent Tutoring Systems (ITS) 2012* (pp. 256–261). Berlin: Springer.

Pashler, H., Cepeda, J. T., Wixted, J. T., & Rohrer, D. (2005). When does feedback facilitate learning of words? *Journal of Experimental Psychology: Learning, Memory, & Cognition, 31*, 3–8.

Person, N. K., Erkel, M., Graesser, A. C., & the Tutoring Research Group. (2002). AutoTutor passes the bystander Turing test. In M. Driscoll & T. C. Reeves (Eds.), *Proceedings for E-Learning 2002: World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education* (pp. 778–782). Montreal: AACE.

Person, N. K., Graesser, A. C., Kreuz, R. J., & Pomeroy, V. (2003). Simulating human tutor dialog moves in AutoTutor. *International Journal of Artificial Intelligence in Education, 12*, 23–39.

Person, N. K., Graesser, A. C., Magliano, J. P., & Kreuz, R. J. (1994). Inferring what the student knows in one-to-one tutoring: the role of student questions and answers. *Learning and Individual Differences, 6*, 205–219.

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

Person, N. K., Olney, A., D'Mello, S. K., & Lehman, B. (2012). Interactive concept maps and learning outcomes in Guru. In G. Youngblood & P. McCarthy (Eds.), *Proceedings of the Florida Artificial Intelligence Research Symposium (FLAIRS) 2012* (pp. 456–461). Menlo Park: AAAI Press.

Prince, M. (2004). Does active learning work? A review of the research. *Journal of Engineering Education, 93* (3), 223–231.

Ritter, S., Anderson, J. R., Koedinger, K. R., & Corbett, A. (2007). Cognitive Tutor: applied research in mathematics education. *Psychonomic Bulletin & Review, 14*(2), 249–255.

Roll, I., Aleven, V., McLaren, B. M., & Koedinger, K. R. (2011). Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system. *Learning and Instruction, 21*(2), 267–280.

Roscoe, R. D., & McNamara, D. S. (2013). Writing pal: feasibility of an intelligent writing strategy tutor in the high school classroom. *Journal of Educational Psychology, 105*(4), 1010–1025.

Rosenshine, B., Meister, C., & Chapman, S. (1996). Teaching students to generate questions: a review of the intervention studies. *Review of Educational Research, 66*, 181–221.

Rouet, J. (2006). *The skills of document use: From text comprehension to web-based learning*. Mahwah: Erlbaum.

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

Rus, V., Baggett, W., Gire, E., Franceschetti, D., Conley, M., Graesser, A.C. (2013a). Towards Learner Models based on Learning Progressions in DeepTutor. In Sottilare, R. (Eds.), *Learner Models*, (pp. 185–196) Army Research Lab.

Rus, V., Banjade, R., Lintean, M., Niraula, N., & Stefanescu, D. (2013b). SEMILAR: A Semantic Similarity Toolkit for Assessing Students' Natural Language Inputs. In D'Mello, S. K., Calvo, R. A., & Olney, A. (Eds.), *Proceedings of Educational Data Mining 2013* (pp. 402–403).

Rus, V., D'Mello, S. K., Hu, X., & Graesser, A. C. (2013c). Recent Advances in conversational intelligent tutoring systems. *AI Magazine, 34*, 42–54.

Rus, V., McCarthy, P. M., McNamara, D. S., & Graesser, A. C. (2008). A study of textual entailment. *International Journal on Artificial Intelligence Tools, 17*(04), 659–685.

Rus, V., McCarthy, P. M., Graesser, A. C., & McNamara, D. S. (2009). Identification of sentence-to-sentence relations using a textual entailer. *Research on Language and Computation, 7*(2–4), 209–229.

Rus, V., Stefanescu, D., Baggett, W., Niraula, N., Franceschetti, D., & Graesser, A. C. (2014). Macro-adaptation in conversational intelligent tutoring matters. In S. Trausan-Matu, K. E. Boyer, M. Crosby, & K. Panourgia (Eds.), *Proceedings of Intelligent Tutoring Systems (ITS) 2014* (pp. 242–247). Berlin: Springer.

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

Self, J. (1990). Theoretical foundations for intelligent tutoring systems. *Journal of Artificial Intelligence in Education, 1*(4), 3–14.

Shaffer, D. W. (2006). *How computer games help children learn*. New York: Palgrave Macmillan.

Shaffer, D. W. & Graesser, A. C. (2010). Using a quantitative model of participation in a community of practice to direct automated mentoring in an ill-defined domain. In C. Lynch, K. Ashley, T. Mitrovic, V. Dimitrova, N. Pinkwart, & V. Aleven (Eds.), *Proceedings of the 4th International Workshop on Intelligent Tutoring Systems and Ill-Defined Domains* (pp. 61–68).

Shubeck, K., Craig, S. D., Hu, X., Faghihi, U., Levy, M., & Koch, R. (2012). Incorporating natural language tutoring into a virtual world for emergency response training. In P. M. McCarthy & G. M. Youngblood (Eds.), *Proceedings of Florida Artificial Intelligence Research Society (FLAIRS) 2012* (p. 573). Menlo Park: AAAI Press.

Shute, V. J. (2008). Focus on formative feedback. *Review of Educational Research, 78*, 153–189.

Song, K. S., Hu, X., Olney, A., & Graesser, A. C. (2004). A framework of synthesizing tutoring conversation capability with web-based distance education courseware. *Computers & Education, 42*(4), 375–388.

Sottilare, R. A., Goldberg, B. S., Brawner, K. W., & Holden, H. K. (2012). A modular framework to support the authoring and assessment of adaptive computer-based tutoring systems (CBTS). In *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC) 2012* (1). Arlington, VA: National Training Systems Association.

Stevens, A. L., & Collins, A. (1977). The goal structure of a Socratic tutor. In J. Ketchel (Ed.), *Proceedings of the ACM Conference 1977* (pp. 256–263). New York: ACM Press.

Susarla, S., Adcock, A., Van Eck, R., Moreno, K., & Graesser, A. C. (2003). Development and evaluation of a lesson authoring tool for AutoTutor. In V. Aleven, U. Hoppe, J. Kay, R. Mizoguchi, H. Pain, F. Verdejo, & K. Yacef (Eds.), *AIED2003 Supplemental Proceedings* (pp. 378–387). Sydney: University of Sydney School of Information Technologies.

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

VanLehn, K., Graesser, A. C., Jackson, G. T., Jordan, P., Olney, A., & Rosé, C. P. (2007). When are tutorial dialogues more effective than reading? *Cognitive Science, 31*(1), 3–62.

VanLehn, K., Siler, S., Murray, C., Yamauchi, T., & Baggett, W. B. (2003). Why do only some events cause learning during human tutoring? *Cognition and Instruction, 21*(3), 209–249.

VanLehn, K., van de Sande, B., Shelby, R., & Gershman, S. (2010). The Andes physics tutoring system: An experiment in freedom. In R. Nkambou, R. Mizoguchi, & J. Bourdeau (Eds.), *Advances in intelligent tutoring systems* (pp. 421–443). Berlin: Springer.

Wang, N., Johnson, W. L., Mayer, R. E., Rizzo, P., Shaw, E., & Collins, H. (2008). The politeness effect: pedagogical agents and learning outcomes. *International Journal of Human-Computer Studies, 66*(2), 98–112.

Wolfe, C. R., Fisher, C. R., Reyna, V. F., & Hu, X. (2012). Improving internal consistency in conditional probability estimation with an intelligent tutoring system and web-based tutorials. *International Journal of Internet Science, 7*(1), 37–54.

Wolfe, C. R., Widmer, C. L., Reyna, V. F., Hu, X., Cedillos, E. M., Fisher, C. R., & Weil, A. M. (2013). The development and analysis of tutorial dialogues in AutoTutor Lite. *Behavior Research Methods, 45*(3), 623–636.

Wood, D., & Wood, H. (1996). Vygotsky, tutoring and learning. *Oxford Review of Education, 22*(1), 5–16.

Woolf, B. P. (2009). *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Burlington: Morgan Kaufmann.

Woolf, B. P., Arroyo, I., Muldner, K., Burleson, W., Cooper, D. G., Dolan, R., & Christopherson, R. M. (2010). The effect of motivational learning companions on low achieving students and students with disabilities. In V. Aleven, J. Kay, & J. Mostow (Eds.), *Proceedings of Intelligent Tutoring Systems (ITS) 2010* (pp. 327–337). Berlin: Springer.