

Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments

W. Lewis Johnson and Jeff W. Rickel
Center for Advanced Research in Technology for Education (CARTE)
USC Information Sciences Institute
4676 Admiralty Way, Marina del Rey, CA 90292 USA
johnson@isi.edu, rickel@isi.edu
<http://www.isi.edu/isd/carte>

James C. Lester
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-7534 USA
lester@csc.ncsu.edu
<http://multimedia.ncsu.edu/imedia>

July 15, 1999

Abstract

Recent years have witnessed the birth of a new paradigm for learning environments: animated pedagogical agents. These lifelike autonomous characters cohabit learning environments with students to create rich, face-to-face learning interactions. This opens up exciting new possibilities; for example, agents can demonstrate complex tasks, employ locomotion and gesture to focus students' attention on the most salient aspect of the task at hand, and convey emotional responses to the tutorial situation. Animated pedagogical agents offer great promise for broadening the bandwidth of tutorial communication and increasing learning environments' ability to engage and motivate students. This article sets forth the motivations behind animated pedagogical agents, describes the key capabilities they offer, and discusses the technical issues they raise. The discussion is illustrated with descriptions of a number of animated agents that represent the current state of the art.

1 Introduction and Background

This paper explores a new paradigm for education and training: face-to-face interaction with intelligent, animated agents in interactive learning environments. The paradigm joins two previously distinct research areas. The first area, animated interface agents (André & Rist 1996; André 1997; Ball *et al.* 1997; Hayes-Roth & Doyle 1998; Laurel 1990; Maes 1994; Nagao & Takeuchi 1994; Thorisson 1996), provides a new metaphor for human-computer interaction based on face-to-face dialogue. The second area, knowledge-based learning environments (Carbonell 1970; Sleeman & Brown 1982; Wenger 1987), seeks instructional

software that can adapt to individual learners through the use of artificial intelligence. By combining these two ideas, we arrive at a new breed of software agent: an *animated pedagogical agent* (Lester *et al.* 1999a; Lester, Stone, & Stelling 1999; Rickel & Johnson 1999a; Shaw, Johnson, & Ganeshan 1999).

Animated pedagogical agents share deep intellectual roots with previous work on knowledge-based learning environments, but they open up exciting new possibilities. As in previous work, students can learn and practice skills in a virtual world, and the computer can interact with students through mixed-initiative, tutorial dialogue (Carbonell 1970) in the role of a coach (Goldstein 1976; Burton & Brown 1982) or learning companion (Chan 1996). However, the vast majority of work on tutorial and task-oriented dialogues has focused on verbal interactions, even though the earliest studies clearly showed the ubiquity of nonverbal communication in similar human dialogues (Deutsch 1974). An animated agent that cohabits the learning environment with students allows us to exploit such nonverbal communication. The agent can demonstrate how to perform actions (Rickel & Johnson 1997a). It can use locomotion, gaze, and gestures to focus the student's attention (Lester *et al.* 1999a; Noma & Badler 1997; Rickel & Johnson 1997a). It can use gaze to regulate turn-taking in a mixed-initiative dialogue (Cassell *et al.* 1994a). Head nods and facial expressions can provide unobtrusive feedback on the student's utterances and actions without unnecessarily disrupting the student's train of thought. All of these nonverbal devices are a natural component of human dialogues. Moreover, the mere presence of a lifelike agent may increase the student's arousal and motivation to perform the task well (Lester *et al.* 1997a; Walker, Sproull, & Subramani 1994). Thus, animated pedagogical agents present two key advantages over earlier work: they increase the bandwidth of communication between students and computers, and they increase the computer's ability to engage and motivate students.

Animated pedagogical agents share aspects in common with synthetic agents developed for entertainment applications (Elliott & Brzezinski 1998): they need to give the user an impression of being lifelike and believable, producing behavior that appears to the user as natural and appropriate. There are two important reasons for making pedagogical agents lifelike and believable. First, lifelike agents are likely to be more engaging, making the learning experience more enjoyable. Second, unnatural behaviors typically call attention to themselves and distract users. As Bates *et al.* (Bates, Loyall, & Reilly 1992) have argued, it is not always necessary for an agent to have deep knowledge of a domain in order for it to generate behavior that is believable. To some extent the same is true for pedagogical agents. We frequently find it useful to give our agents behaviors that make them appear knowledgeable, attentive, helpful, concerned, etc. These behaviors may or may not reflect actual knowledge representations and mental states and attitudes in the agents. However, the need to support pedagogical interactions generally imposes a closer correspondence between appearance and internal state than is typical in agents for entertainment applications. We can create animations that give the impression that the agent is knowledgeable, but if the agent is unable to answer student questions and give explanations, the impression of knowledge will be quickly destroyed.

Animated pedagogical agents also share issues with work on autonomous agents, *i.e.*, systems that are capable of performing tasks and achieving goals in complex, dynamic environments. Architectures such as RAP (Firby 1994) and Soar (Laird, Newell, & Rosenbloom 1987) have been used to create agents that can seamlessly integrate planning and execution, adapting to changes in their environments. They are able to interact with other agents and collaborate with them to achieve common goals (Müller 1996; Tambe 1997). Pedagogical-

cal agents must likewise exhibit robust behavior in rich, unpredictable environments; they must coordinate their behavior with that of other agents; and they must manage their own behavior in a coherent fashion, arbitrating between alternative actions and responding to a multitude of environmental stimuli. Their environment includes both students and the learning environment in which the agents are situated. Student behavior is by nature unpredictable, since students may exhibit a variety of aptitudes, levels of proficiency, and learning styles. However, the need to support instruction imposes additional requirements that other types of agents do not always satisfy; in order to support instructional interactions, a pedagogical agent requires a deeper understanding of the rationales and relationships between actions than would be needed simply to perform the task (Clancey 1983).

This paper lays out the motivations behind animated pedagogical agents, the key capabilities they offer, and the technical issues they raise. Full technical accounts of individual methods and systems can be found in the cited references.

2 Example Pedagogical Agents

This paper will make frequent reference to several implemented animated pedagogical agents. These agents will be used to illustrate the range of behaviors that such agents are capable of producing and the design requirements that they must satisfy. Some of these behaviors are similar to those found in intelligent tutoring systems, while others are quite different and unique.

The USC Information Sciences Institute's Center for Advanced Research in Technology for Education (CARTE) has developed two animated pedagogical agents: Steve (Soar Training Expert for Virtual Environments) and Adele (Agent for Distance Learning; Light Edition). Steve (Figure 1) is designed to interact with students in networked immersive virtual environments, and has been applied to naval training tasks such as operating the engines aboard US Navy surface ships (Johnson *et al.* 1998; Johnson & Rickel 1998; Rickel & Johnson 1999a; 1997b). Immersive virtual environments permit rich interactions between humans and agents; the students can see the agents in stereoscopic 3D and hear them speak, and the agents rely on the virtual environment's tracking hardware to monitor the student's position and orientation in the environment. Steve is combined with 3D display and interaction software by Lockheed Martin (Stiles, McCarthy, & Pontecorvo 1995), simulation authoring software by USC Behavioral Technologies Laboratory (Munro *et al.* 1993), and speech recognition and generation software by Entropic Research to produce a rich virtual environment in which students and agents can interact in instructional settings.

Adele (Figure 2), in contrast, was designed to run on desktop platforms with conventional interfaces, in order to broaden the applicability of pedagogical agent technology. Adele runs in a student's Web browser and is designed to integrate into Web-based electronic learning materials (Shaw, Johnson, & Ganeshan 1999; Shaw *et al.* 1999). Adele-based courses are currently being developed for continuing medical education in family medicine and graduate level geriatric dentistry, and further courses are planned for development both at the University of Southern California and at the University of Oregon.

North Carolina State University's IntelliMedia Initiative has developed three animated pedagogical agents: Herman the Bug (Lester, Stone, & Stelling 1999), Cosmo (Lester *et al.* 1999a), and WhizLow (Lester *et al.* 1999b). Herman the Bug inhabits Design-A-Plant, a learning environment for the domain of botanical anatomy and physiology (Figure 3). Given a set of environmental conditions, children interact with Design-A-Plant by graphically as-

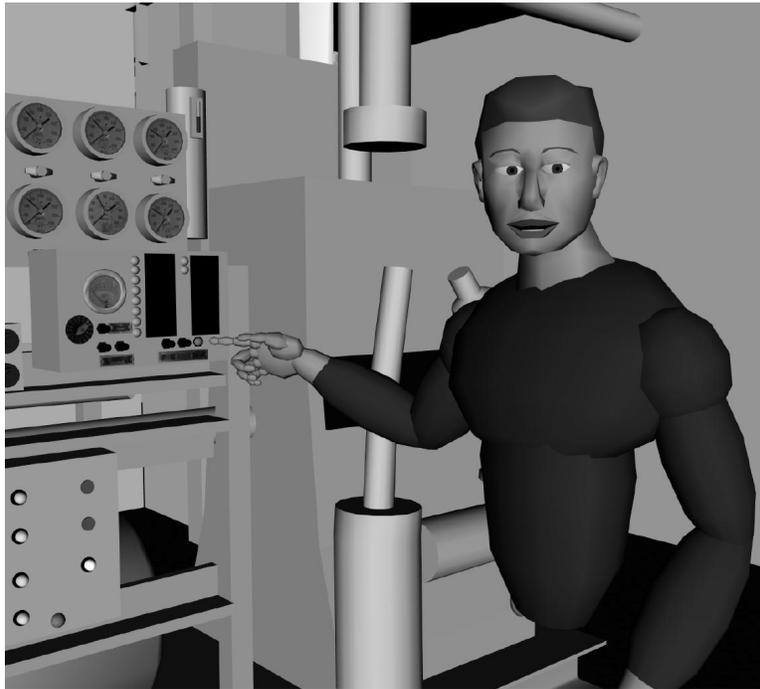


Figure 1: Steve

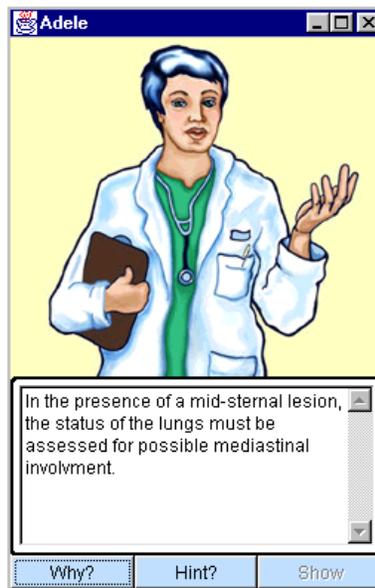


Figure 2: Adele



Figure 3: Herman the Bug

sembling customized plants that can thrive in those conditions. Herman is a talkative, quirky insect that dives into plant structures as he provides problem-solving advice to students. As students build plants, Herman observes their actions and provides explanations and hints. In the process of explaining concepts, he performs a broad range of actions, including walking, flying, shrinking, expanding, swimming, fishing, bungee jumping, teleporting, and acrobatics.

Cosmo provides problem-solving advice in the Internet Protocol Advisor (Figure 4). Students interact with Cosmo as they learn about network routing mechanisms by navigating through a series of subnets. Given a packet to escort through the Internet, they direct it through networks of connected routers. At each subnet, they may send their packet to a specified router and view adjacent routers. By making decisions about factors such as address resolution and traffic congestion, they learn the fundamentals of network topology and routing mechanisms. Helpful, encouraging, and with a bit of an attitude, Cosmo explains how computers are connected, how routing is performed, and how traffic considerations come into play. Cosmo was designed to study spatial deixis in pedagogical agents, i.e., the ability of agents to dynamically combine gesture, locomotion, and speech to refer to objects in the environment while they deliver problem-solving advice.

The WhizLow agent inhabits the CPU City 3D learning environment (Figure 5). CPU City's 3D world represents a motherboard housing three principal components: the RAM, the CPU, and the hard drive. It focuses on architecture including the control unit (which is reduced to a simple decoder) and an ALU, system algorithms such as the fetch cycle, page faults, and virtual memory, and the basics of compilation and assembly. WhizLow can carry out students' tasks by picking up data and instruction packets, dropping them off in specified locations such as registers, and interacting with devices that cause arithmetic and comparison operations to be performed. He manipulates address and data packets, which can contain integer-valued variables. As soon as task specification is complete, he begins performing the student's task in less than one second.

André, Rist, and Müller at DFKI (the German Research Center for Artificial Intelli-

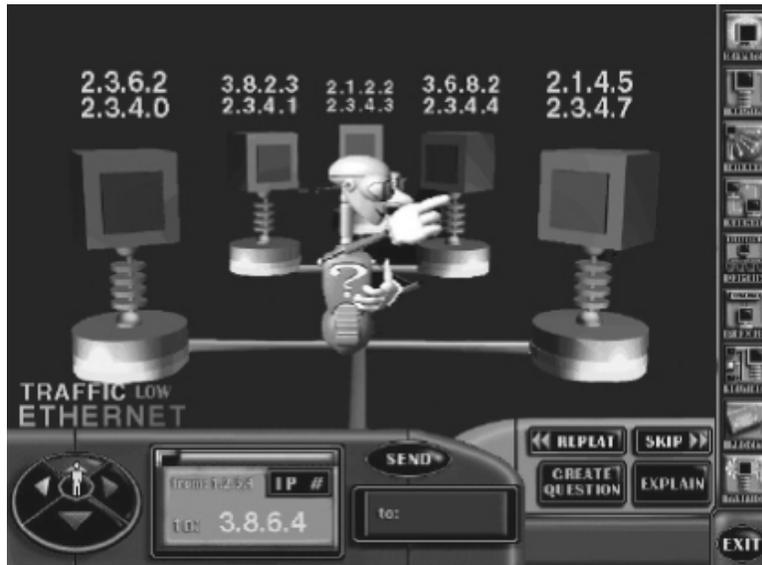


Figure 4: Cosmo

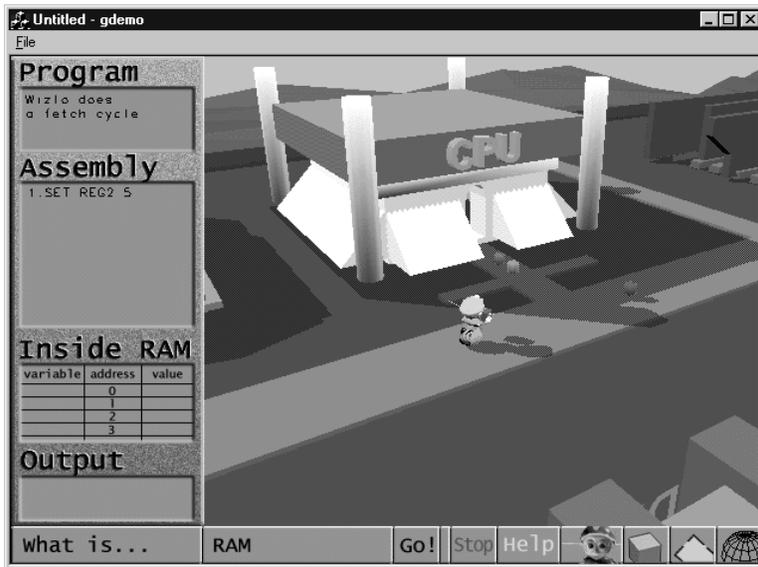


Figure 5: WhizLow

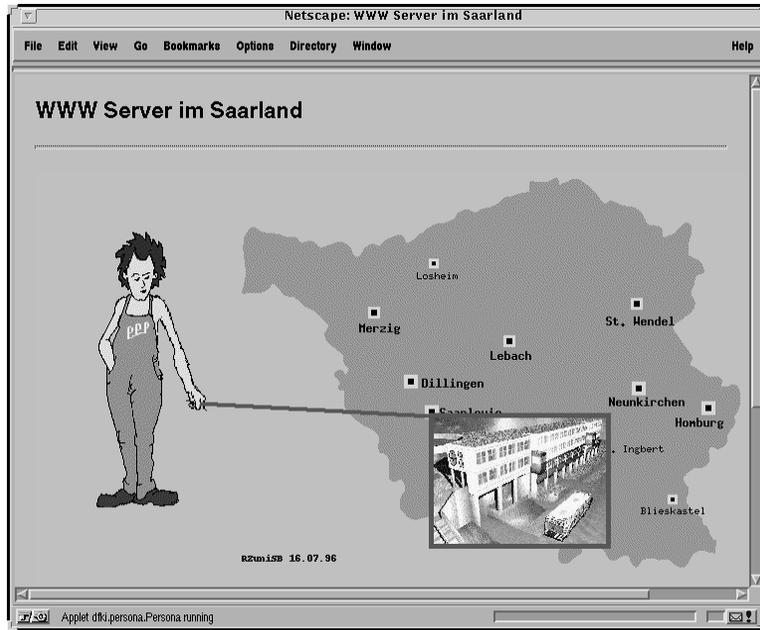


Figure 6: PPP Persona

gence) have developed an animated agent for giving on-line help instructions, called the PPP Persona (André, Rist, & Müller 1999). The agent guides the learner through Web-based materials, using pointing gestures to draw the student's attention to elements of Web pages, and providing commentary via synthesized speech (Figure 6). The underlying PPP system generates multimedia presentation plans for the agent to present; the agent then executes the plan adaptively, modifying it in real time based on user actions such as repositioning the agent on the screen or asking follow-on questions.

3 Enhancing Learning Environments with Animated Agents

This section lists the key benefits provided by animated pedagogical agents by describing the novel types of human-computer interaction they support. No current agent supports all of these types of interaction. Each type can significantly enhance a learning environment without the others, and different combinations will be useful for different kinds of learning environments. To provide a summary of achievements to date, we use existing agents to illustrate each type of interaction. At the end of the section, we discuss some early empirical results on the effectiveness of animated pedagogical agents.

3.1 Interactive Demonstrations

A simulated mock-up of a student's real work environment, coupled with an animated agent that inhabits the virtual world, provides new opportunities for teaching the student how to perform tasks in that environment. Perhaps the most compelling advantage is that the agent can demonstrate physical tasks, such as operation and repair of equipment. For example, Figures 1 and 7 depict Steve showing a student how to operate a High Pressure Air Compressor (HPAC) aboard a US Navy ship. Steve integrates his demonstrations with



Figure 7: Steve pressing a button on the HPAC console

spoken commentary describing objectives and actions. Figure 1 shows him providing such commentary:

I will now perform a functional check of the temperature monitor to make sure that all of the alarm lights are functional. First, press the function test button. This will trip all of the alarm switches, so all of the alarm lights should illuminate.

Steve then proceeds with the demonstration, as shown in Figure 7. As the demonstration proceeds, Steve points out important features of the objects in the environment that relate to the task. For example, when the alarm lights illuminate, Steve points to the lights and says “All of the alarm lights are illuminated, so they are all working properly.”

Demonstrating a task may be far more effective than trying to describe how to perform it, especially when the task involves spatial motor skills, and the experience of seeing a task performed is likely to lead to better retention. Moreover, an interactive demonstration given by an agent offers a number of advantages over showing students a videotape. Students are free to move around in the environment and view the demonstration from different perspectives. They can interrupt with questions, or even ask to finish the task themselves, in which case Steve will monitor the student’s performance and provide assistance. Also, Steve is able to construct and revise plans for completing a task, so he can adapt the demonstration to unexpected events. This allows him to demonstrate the task under different initial states and failure modes, as well as help the student recover from errors.

The utility of agent demonstrations is not restricted to teaching physical tasks that the student must perform. Agents can also demonstrate procedures performed by complex devices by taking on the role of an actor in a virtual process. For example, WhizLow, the agent in the CPU City learning environment, demonstrates computational procedures to

teach novices the fundamentals of computer architecture. As he transports data packets and addresses packets to the CPU, RAM, and hard drive, WhizLow teaches students how fetch-execute cycle algorithms work. In contrast to Steamer-style interactions (Hollan, Hutchins, & Weitzman 1984; Stevens, Roberts, & Stead 1983) in which knowledge-based simulations guide the actions in a simulated world, learning environments in which the instructions are provided by lifelike characters provide a visual focus and an engaging presence that are sometimes absent from their agentless counterparts.

3.2 Navigational Guidance

When a student's work environment is large and complex, such as a ship, one of the primary advantages of a virtual mock-up is to teach the student where things are and how to get around. In this context, animated agents are valuable as navigational guides, leading students around and preventing them from becoming lost. For example, Steve inhabits a complex shipboard environment, including multiple rooms. The engine room alone is quite complex, with the large turbine engines that propel the ship, several platforms and pathways around and into the engines, a console, and a variety of different parts of the engines that must be manipulated, such as valves. As Steve demonstrates tasks, he leads students around this environment, showing them where relevant objects are and how to get to them. Because Steve has an internal representation of the spatial layout of the ship (see Section 4), he is always able to plan the shortest path from his current location to the next relevant object. Leading someone down a hallway, up a flight of stairs, around a corner, and through some pipes to the valve they must turn is likely to be more effective than trying to tell them where the valve is located. Our experience in training people using immersive virtual reality has shown that students can easily become disoriented and lost in complex environments, so animated agents that can serve as guides are an important instructional aid.

By enabling students to participate in immersive experiences, 3D learning environments with navigational guides can help students develop spatial models of the subject matter, even if these environments present worlds that the student will never occupy. For example, the CPU City environment depicts a virtual computer that the student can travel through and interact with to acquire a mental model of the workings of a computer. Similar experiences could be provided by learning environments that offer students tours of civilizations long past, e.g., the wonders of ancient Greece, or of virtual museums housing the world's masterpieces. Accompanied by knowledgeable guides, students can travel through these virtual worlds to learn about a variety of domains that lend themselves to spatial exploratory metaphors.

Although Steve and WhizLow both inhabit 3D worlds, an animated navigational guide may even be useful in 2D environments. For example, the CAETI Center Associate (Murray 1997) serves as a Web-based guide to a large collection of intelligent tutoring system projects. A virtual building houses these projects in individual "rooms." When a user first enters the world, the CAETI guide interviews her about her interests to construct a customized itinerary. It then escorts her from room to room (project to project) based on her interests. While the guides described above help students navigate 3D worlds, the CAETI Associate demonstrates that 2D worlds may also benefit from the presence of animated agents.

3.3 Gaze and Gesture as Attentional Guides

Because of significant advances in the capabilities of graphics technologies in the past decade, tutoring systems increasingly incorporate visual aids. These range from simple maps or charts that are automatically generated (Mittal *et al.* 1995) to 3D simulations of physical phenomena such as electromagnetic interactions in physics (Towns, Callaway, & Lester 1998) and fullscale 3D simulated worlds such as the ship that Steve inhabits. To draw students' attention to a specific aspect of a chart, graphic or animation, tutoring systems make use of many devices, such as arrows and highlighting by color. An animated agent, however, can guide a student's attention with the most common and natural methods: gaze and deictic gesture.

Steve uses gaze and deictic gestures in a variety of ways. He points at objects when discussing them. He looks at an object immediately before manipulating or pointing at it. He looks at objects when they are manipulated by students or other agents. He looks at an object when checking its state (e.g., to see whether a light is on or a reservoir is full). He looks at a student or another agent when waiting for them, listening to them, or speaking to them. Steve is even capable of tracking moving objects; for example, if something (e.g., the student) is moving counterclockwise around Steve, he will track it over his left shoulder until it moves directly behind him, at which point he will track it over his right shoulder.

Agents can employ deictic behaviors to create context-specific references to physical objects in virtual worlds. In the same manner that humans refer to objects in their environment through judicious combinations of speech, locomotion, and gesture, animated agents can move through their environment, point to objects, and refer to them appropriately as they provide problem-solving advice. An agent might include some or all of these capabilities. For example, to produce deictic references to particular objects under discussion, the Edward system (Claassen 1992) employs a stationary persona that "grows" a pointer to a particular object in the interface. Similarly, the PPP Persona is able to dynamically indicate various onscreen objects with an adjustable pointer (Figure 6). Adele is able to point toward objects on the screen, and can also direct her gaze toward them; Figure 8 shows her looking at the student's mouse selection. The Cosmo agent employs a deictic behavior planner that exploits a simple spatial model to select and coordinate locomotive, gestural, and speech behaviors. The planner enables Cosmo to walk to, point at, and linguistically refer to particular computers in its virtual world as it provides students with problem-solving advice.

Noma and Badler's Presenter Jack (Noma & Badler 1997), shown in Figure 9, exhibits a variety of different deictic gestures. Like Steve and Cosmo, Presenter Jack can use his index finger to point at individual elements on his visual aid. He can also point with his palm facing towards the visual aid to indicate a larger area, and he can move his hand to indicate a flow on a map or chart. He also smoothly integrates these gestures into his presentation, moving over to the target object before his speech reaches the need for the deictic gesture, and dynamically choosing the best hand for the gesture based on a heuristic that minimizes both visual aid occlusion and the distance from the current body position to the next one in the presentation.

3.4 Nonverbal Feedback

One primary role of a tutor is to provide feedback on a student's actions. In addition to providing verbal feedback, an animated agent can also use nonverbal communication

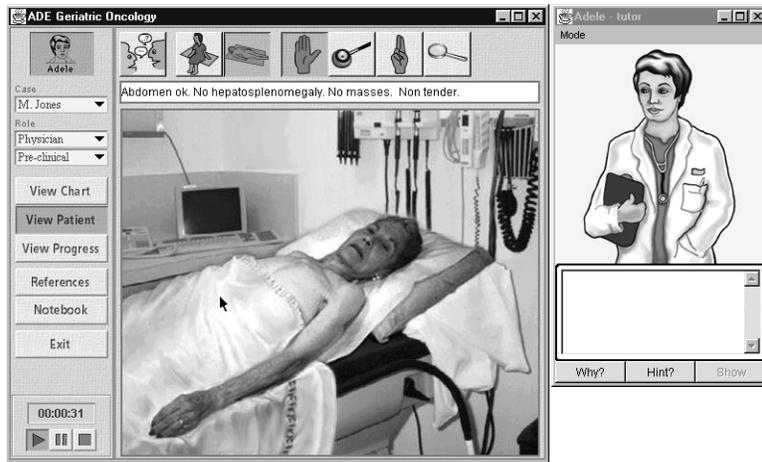


Figure 8: Adele looking at the student's mouse selection

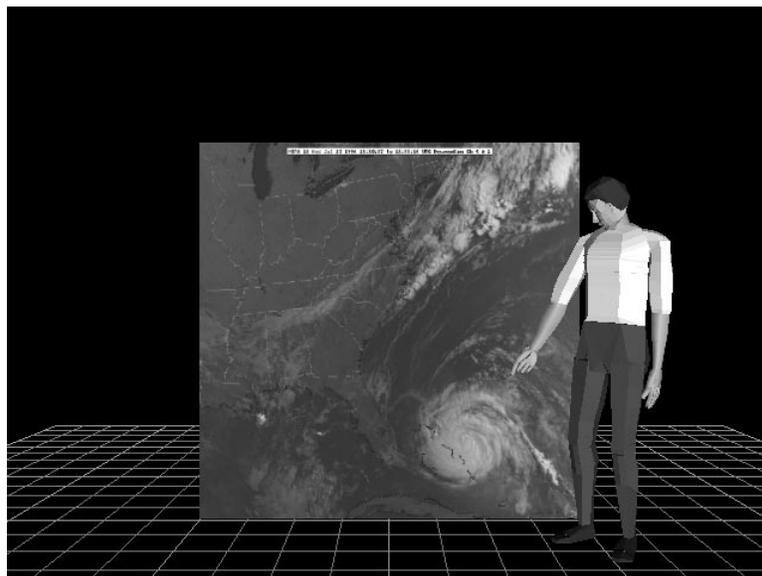


Figure 9: Presenter Jack pointing at a weather pattern

to influence the student. For example, Steve uses a nod of approval to show agreement with a student's actions and shakes his head to indicate disapproval. Adele nods or smiles to indicate agreement with the student's actions, presents a look of puzzlement when the student makes an error, and shows pleasant surprise when the student finishes their task. Moreover, body language can help indicate to students that they have just committed (or are on the verge of committing) a very serious error. This can make a strong impression on them.

The ability to use nonverbal feedback in addition to verbal comments allows an animated agent to provide more varied degrees of feedback than earlier tutoring systems. Nonverbal feedback through facial expressions may often be preferable because it is less obtrusive than a verbal comment. For example, a simple nod of approval can reassure a student without interrupting them. Similarly, human tutors often display a look of concern or puzzlement to make a student think twice about their actions in cases where either they are unsure that the student has actually made a mistake or they do not want to interrupt with a verbal correction yet. While some occasions call for these types of unobtrusive feedback, other occasions may call for more exaggerated feedback than a verbal comment can offer. For example, when students successfully complete design problems in the Design-A-Plant learning environment, the animated agent (Herman) sometimes congratulates them by cartwheeling across the screen. In the Internet Advisor, Cosmo employs "stylized" animations (Culhane 1988) (in contrast to "life-quality" animations) for nonverbal feedback. For example, when a student solves a problem, Cosmo smiles broadly and uses his entire body to applaud her success.

3.5 Conversational Signals

When people carry on face-to-face dialogues, they employ a wide variety of nonverbal signals to help regulate the conversation and complement their verbal utterances. While tutorial dialogue in most previous tutoring systems resembles Internet chat or a phone conversation, animated pedagogical agents allow us to more closely model the face-to-face interactions to which people are most accustomed. Some nonverbal signals are closely tied to spoken utterances, and could be used by any animated agent that produces speech output. For example, intonational pitch accents indicate the degree and type of salience of words and phrases in an utterance, including rhematic (i.e., new) elements of utterances and contrastive elements (Pierrehumbert & Hirschberg 1990); to further highlight such utterance elements, a pitch accent is often accompanied by a short movement of the eyebrows or head, a blink of the eyes, and/or a beat gesture (i.e., a short baton-like movement of the hands) (Cassell *et al.* 1994a). As another example, facial displays can provide the speaker's personal judgement of the accompanying utterance (e.g., a scrunched nose to indicate distaste for the subject) (Cassell *et al.* 1994a).

Other nonverbal signals help regulate the flow of conversation, and would be most valuable in tutoring systems that support speech recognition as well as speech output, such as Steve or the Circuit Fix-It Shop (Smith & Hipp 1994). This includes back-channel feedback, such as head nods to acknowledge understanding of a spoken utterance. It also includes the use of eye contact to regulate turn taking in mixed-initiative dialogue. For example, during a pause, a speaker will either break eye contact to retain the floor or make eye contact to request feedback or give up the floor (Cassell *et al.* 1994a). Although people can clearly communicate in the absence of these nonverbal signals (e.g., by telephone), communication and collaboration proceed most smoothly when they are available.

Several projects have made serious attempts to draw on the extensive psychological

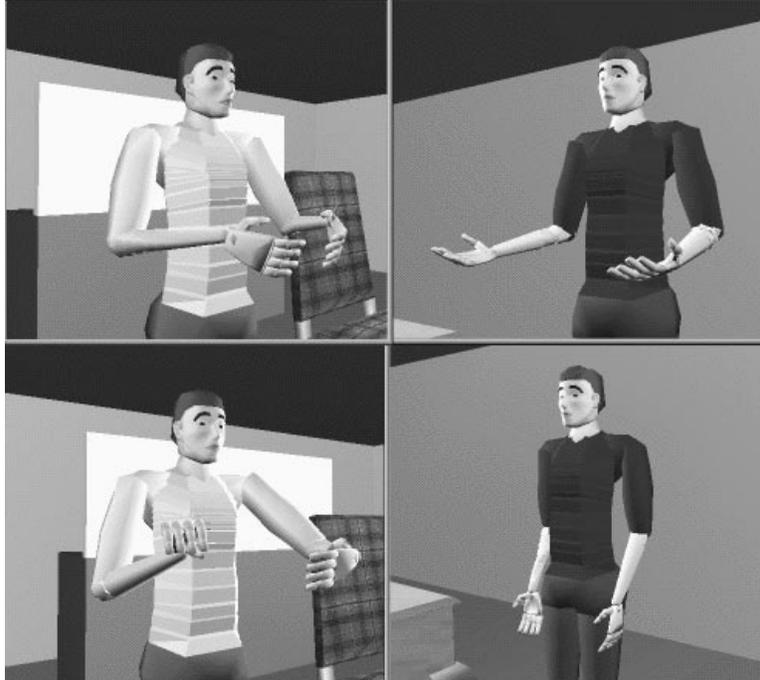


Figure 10: Animated Conversation

and sociological literature on human nonverbal conversational behavior. Pelachaud et al. (Pelachaud, Badler, & Steedman 1996) developed a computational model of facial expressions and head movements of a speaker. Cassell et al. (Cassell *et al.* 1994a; 1994b) developed perhaps the most comprehensive computational model of nonverbal communicative behavior. Their agents coordinate speech, intonation, gaze, facial expressions, and a variety of gestures in the context of a simple dialogue. However, their agents do not converse with humans; their algorithm simply generates an animation file for a face-to-face conversation between two computer characters, Gilbert and George (Figure 10), using the Jack human figure software (Badler, Phillips, & Webber 1993). In contrast, the Gandalf agent (Figure 11) supports full multi-modal conversation between human and computer (Thorisson 1996; Cassell & Thorisson 1999). Like other systems, Gandalf combines speech, intonation, gaze, facial expressions, and a few gestures. Unlike most other systems, Gandalf also perceives these communicative signals in humans; people talking with Gandalf wear a suit that tracks their upper body movement, an eye tracker that tracks their gaze, and a microphone that allows Gandalf to hear their words and intonation. Although none of these projects has specifically addressed tutorial dialogues, they contribute significantly to our understanding of communication with animated agents.

3.6 Conveying and Eliciting Emotion

Motivation is a key ingredient in learning, and emotions play an important role in motivation. By employing a computational model of emotion, animated agents can improve students' learning experiences in several ways (Elliott, Rickel, & Lester 1999). First, an agent that appears to care about a student's progress may encourage the student to care more about her own progress. Second, an emotive pedagogical agent may convey enthusiasm for the subject matter and thereby foster similar levels of enthusiasm in the learner.



Figure 11: Gandalf speaking with a user

Finally, a pedagogical agent with a rich and interesting personality may simply make learning more fun. A learner that enjoys interacting with a pedagogical agent may have a more positive perception of the overall learning experience and may consequently opt to spend more time in the learning environment.

Perhaps as a result of the inherent psychosocial nature of student-agent interactions and of humans' tendency to anthropomorphize software (Reeves & Nass 1998), recent evidence suggests that tutoring systems with lifelike characters can be pedagogically effective (Lester *et al.* 1997b) while at the same time having a strong motivating effect on students (Lester *et al.* 1997a). It is even becoming apparent that particular features (e.g., personal characteristics) of lifelike agents can have an important impact on learners' acceptance of them (Hietala & Niemirepo 1998). As master animators have discovered repeatedly over the past century, the quality, overall clarity, and dramatic impact of communication can be increased through the creation of emotive movement that underscores the affective content of the message to be communicated (Noake 1988; Jones 1989; Lenburg 1993; Thomas & Johnston 1981). By carefully orchestrating facial expression, body placement, arm movements, and hand gestures, animated pedagogical agents could visually augment verbal problem-solving advice, give encouragement, convey empathy, and perhaps increase motivation. For example, the Cosmo agent employs a repertoire of "full-body" emotive behaviors to advise, encourage, and (appear to) empathize with students. When a student makes a sub-optimal problem-solving decision, Cosmo informs the student of the ill-effect of her decision as he takes on a sad facial expression and slumping body language while dropping his hands. As computational models of emotion become more sophisticated, e.g., (Elliott 1992), animated agents will be well positioned to improve students' motivation.

3.7 Virtual Teammates

Complex tasks often require the coordinated actions of multiple team members. Team tasks are ubiquitous in today's society; for example, teamwork is critical in manufacturing, in an emergency room, and on a battlefield. To perform effectively in a team, each member

must master their individual role and learn to coordinate their actions with their teammates. Distributed virtual reality provides a promising vehicle for training teams; students, possibly at different locations, cohabit a virtual mock-up of their work environment, where they can practice together in realistic situations. In such training, animated agents can play two valuable roles: they can serve as instructors for individual students, and they can substitute for missing team members, allowing students to practice team tasks when some or all human instructors and teammates are unavailable.

Steve supports this type of training (Rickel & Johnson 1999b). The team can consist of any combination of Steve agents and human students, each assigned a particular role in the team (e.g., officer of the watch or propulsion operator). Each student is accompanied by an instructor (human or agent) that coaches them on their role. Each person sees each other person in the virtual world as a head and two hands; the head is simply a graphical model, so each person can have a distinct appearance, possibly with their own face texture-mapped onto the graphical head. To distinguish different agents, each agent can be configured with its own shirt, hair, eye, and skin color, and its voice can be made distinct by setting its speech rate, base-line pitch, and vocal tract size. Thus, students can easily track the activities of their teammates. Team members communicate through spoken dialogue, and Steve agents also incorporate valuable nonverbal communication: they look at a teammate when waiting for them or speaking to them, they react to their teammates' actions, and they nod in acknowledgment when they understand something a teammate says to them. Each Steve agent's behavior is guided by a task representation that specifies the overall steps in the task as well as how various team members interact and depend upon each other.

In addition to serving as teammates, animated pedagogical agents could serve as other types of companions for students. Chan and Baskin (Chan & Baskin 1990) developed a simulated learning companion, which acts as a peer instead of a teacher. Dillenbourg (Dillenbourg 1996) investigated the interaction between real students and computer-simulated students as a collaborative social process. Chan (Chan 1996) has investigated other types of interactions between students and computer systems, such as competitors or reciprocal tutors. Frasson *et al.* (Frasson *et al.* 1996) have explored the use of an automated "troublemaker," a learning companion that sometimes provides incorrect information in order to check, and improve, the student's self-confidence. None of these automated companions appears as an animated character, although recent work by Aïmeur *et al.* (Aïmeur *et al.* 1997) has explored the use of a 2D face with facial expressions for the troublemaker. However, since all these efforts share the perspective of learning as a social process, this seems like a natural direction for future research.

3.8 Adaptive Pedagogical Interactions

In addition to the types of interactions described above, animated pedagogical agents need to be capable of many of the same pedagogical abilities as other intelligent tutoring systems. For instance, it is useful for them to be able to answer questions, generate explanations, ask probing questions, and track the learners' skill levels. An animated pedagogical agent must be able to perform these functions while at the same time responding to the learners' actions. Thus the context of face-to-face interaction has a pervasive influence on the pedagogical functions incorporated in an animated pedagogical agent; pedagogy must be dynamic and adaptive, as opposed to deliberate, sequential, or preplanned. For example, Steve adapts his demonstrations in midstream if the student performs actions that interact with the demonstration; he also responds to student interruptions. Similarly, the PPP

Persona seamlessly integrates reactive behaviors responding to user inputs with planned presentations.

The ability to deliver opportunistic instruction, based on the current situation, is a common trait of animated pedagogical agents. Herman the Bug, for example, makes extensive use of problem solving contexts as opportunities for instruction. When the student is working on selecting a leaf to include in a plant, Herman uses this as an opportunity to provide instruction about leaf morphology. Adele constantly assesses the current situation, using the situation space model of Marsella and Johnson (Marsella & Johnson 1998), and dynamically generates advice appropriate to the current situation. Another type of opportunistic instruction provided by Adele is suggesting pointers to on-line medical resources that are relevant to the current stage of the case work-up. For example, when the student selects a diagnostic procedure to perform on the simulated patient, Adele may point the student to video clips showing how the procedure is performed.

3.9 Preliminary Empirical Results

Because animated pedagogical agent technologies are still very much in their infancy, little is known empirically about their effectiveness in learning environments. As discussed in the next section, nearly every major facet of their communicative abilities needs considerable research. For this reason, it is much too early in their development to conduct comprehensive, definitive empirical studies that demonstrate their effectiveness in learning environments. Because their communicative abilities are still very limited compared to what we expect they will be in the near future, the results of such studies will be skewed by the immaturity of the technology. Despite this caveat, it is essential to make an initial foray into assessing their impact on learning, and several studies have been undertaken with this objective in mind. Below we summarize the results of several representative studies.¹

The largest formal empirical study of an animated pedagogical agent to date was conducted with Herman the Bug in the Design-A-Plant learning environment (Lester *et al.* 1997b). Researchers wanted to obtain a “baseline” reading on the potential effectiveness of animated pedagogical agents and examine the impact of various forms of agents’ advice. They conducted a study with one hundred middle school students in which each student interacted with one of several versions of the Herman agent. The different versions varied along two dimensions. First, different versions of Herman employed different modalities: some provided only visual advice, some only verbal advice, and some provided combinations of the two. Second, different versions provided different levels of advice: some agents provided only high-level (principle-based) advice, others provided low-level (task-specific) advice, and some were completely mute. During the interactions, the learning environment logged all problem-solving activities, and the students were given rigorous pre-tests and post-tests. The results of the study were three-fold:

Baseline Result Students interacting with learning environments with an animated pedagogical agent show statistically significant increases from pre-tests to post-tests. Some critics have suggested that animated agents could distract students and hence prevent learning. This finding establishes that a well-designed agent in a well-designed learning environment can create successful learning experiences.

Multi-Level, Multi-Modality Effects Animated pedagogical agents that provide multiple levels of advice combining multiple modalities yield greater improvements in

¹Complete descriptions of the experimental methods and analyses are contained in the cited papers.

problem solving than less expressive agents. This finding indicates that there may be important learning benefits from introducing animated agents that employ both visual (animated) and auditory (verbal) modalities to give both “practical” and “theoretical” advice.

Complexity Benefits The benefits of animated pedagogical agents increase with problem-solving complexity. As students are faced with more complex problems, the positive effects of animated pedagogical agents on problem solving are more pronounced. This finding suggests that agents may be particularly effective in helping students solve complex technical problems (as opposed to simple “toy” problems).

The Design-A-Plant study also revealed the *persona effect* (Lester *et al.* 1997a): the very presence of a lifelike character in an interactive learning environment can have a strong positive effect on learners’ perception of their learning experience. The study also demonstrated an important synergistic effect of multiple types of explanatory behaviors on students’ perception of agents: agents that are more expressive (both in modes of communication and in levels of advice) are perceived as having greater utility and communicating with greater clarity.

In a separate study, the PPP research team conducted an experiment to evaluate the degree to which their PPP agent contributes to learning (André, Rist, & Müller 1999). To this end, they created two versions of their learning environment software, one with the PPP Persona and one without. The latter uses identical narration and uses an arrow for deictic reference. Each subject (all of them adults) viewed several presentations; some presentations provided technical information (descriptions of pulley systems) while others provided non-technical information (descriptions of office employees). Unlike the Design-A-Plant study, the subjects in this study did not perform any problem solving under the guidance of the agent. The results indicate that the presence of the animated agent made no difference to subjects’ comprehension of the presentations. This finding neither supports nor contradicts the Design-A-Plant study, which did not involve an agent vs. no-agent comparison, and which involved a very different learning environment. However, 29 out of 30 subjects in the PPP study preferred the presentations with the agent. Moreover, subjects found the technical presentations (but not the non-technical presentations) significantly less difficult and more entertaining with the agent. This result is consistent with the persona effect found in the Design-A-Plant study.

It is important to emphasize that both of these studies were conducted with agents that employed “first generation” animated pedagogical agent technologies. All of their communicative capabilities were very limited compared to the level of functionality that is expected to emerge over the next few years, and Herman and the PPP Persona only employ a few of the types of interaction that have been discussed in this paper. As animated pedagogical agents become more sophisticated, it will be critical to repeat these experiments en route to a comprehensive, empirically-based theory of animated pedagogical agents and learning effectiveness.

4 Technical Issues

Animated pedagogical agents share many technical issues with previous work in intelligent tutoring systems and interactive learning environments, including representing and reasoning about domain knowledge, modeling and adapting to the student’s knowledge, choosing

appropriate pedagogical strategies, and maintaining a coherent tutorial dialogue. However, just as animated agents raise new instructional opportunities, as described in the last section, they also pose new technical challenges. This section outlines the key challenges and some of the relevant work to date in addressing them.

4.1 Interface to the Environment

Viewed as an autonomous agent, an animated pedagogical agent’s “environment” includes the learning environment (e.g., anything from a 3D virtual world to a simple 2D Web interface), the student(s), and any other agents in the learning environment. Before discussing the inner workings of such an agent, it is helpful to discuss the interface between the agent and this environment. The interface can be divided into two parts: the agent’s awareness of the environment (its “perception”), and its ability to affect the environment (its “motor actions”). One of the primary motivations for animated pedagogical agents is to broaden the bandwidth of human-computer interaction, so their perception and motor actions are typically more diverse than previous computer tutors and learning companions.

Animated pedagogical agents share some types of perception with earlier tutoring systems. Most track the state of the problem the student is addressing. For example, Steve tracks the state of the simulated ship, Adele tracks the state of the simulated patient, and Herman maintains a representation of the environment for which the student is designing a plant. Most track the student’s problem-solving actions. For example, Steve knows when the student manipulates objects (e.g., pushes buttons or turns knobs), Adele knows when the student questions or examines the patient (e.g., inspects a lesion or listens to the heart), and Herman knows when the student extends the plant design (e.g., chooses the type of leaves). Finally, most allow the student to ask them questions. For example, students can ask Steve and Adele what they should do next and why, they can ask Herman and Cosmo for problem-solving assistance, and they can ask WhizLow to perform a task that they have designed for him.

In addition, some agents track other, more unusual events in their environment. Some track additional speech events. When an external speech synthesizer is used to generate the agent’s voice, the agent must receive a message indicating when speech is complete, and the agent may receive interim messages during speech output specifying information such as the appropriate viseme for the current phoneme (for lip synchronization) or the timing of a pitch accent (for coordinated use of a beat gesture, a head movement, or raised eyebrows). To maintain awareness of when others are speaking, the agent may receive messages when the student begins and finishes speaking (e.g., from a speech recognition program) and when other agents begin or finish speaking (from their speech synthesizers), as well as a representation of what was said. Some agents, such as Steve, track the student’s location in the virtual world, and agents for team training may track the locations of other agents. Some track the student’s visual attention. For example, Steve gets messages from the virtual reality software indicating which objects are within the student’s field of view, and he pauses his demonstrations when the student is not looking in the right place. Gandalf tracks the student’s gaze as a guide to conversational turn taking, and he also tracks their gestures. It is very likely that future pedagogical agents will track still other features, such as students’ facial expressions (Cohn *et al.* 1998) and emotions (Picard 1997).

Interactions between an agent’s body and its environment require spatial knowledge of that environment. As described in Section 3, such interactions are a key motivation for animated pedagogical agents, including the ability to look at objects, point at them,

demonstrate how to manipulate them, and navigate around them. Relatively simple representations of spatial knowledge have sufficed to support the needs of animated pedagogical agents to date. For example, Herman maintains a simple representation of the student's "task bar" location in the Design-A-Plant environment so he can conduct his activities (e.g., standing, sitting, walking) appropriately on the screen. Agents such as the PPP Persona that point at elements of bitmapped images need the screen location of the referenced elements. Cosmo maintains a similar representation of the locations of objects on the screen so he can perform his deictic locomotion and gestural behaviors; he also uses this knowledge for selecting appropriate referring expressions.

Agents that inhabit 3D worlds require still richer representations. Steve relies on the virtual reality software to provide bounding spheres for objects, thus giving him knowledge of an object's position and a coarse approximation of its spatial extent for purposes of gaze and deictic gesture. Steve also requires a vector pointing at the front of each object (from which he determines where to stand) and, to support object manipulation, vectors specifying the direction to press or grasp each object. WhizLow maintains knowledge about the physical properties of various objects and devices. For example, the representation encodes knowledge that data packets can be picked up, carried, and deposited in particular types of receptacles and that levers can be pulled.

Agents in 3D environments may need additional knowledge to support collision-free locomotion. Steve represents the world as an adjacency graph: each node in the graph represents a location, and there is an edge between two nodes if there is a collision-free path directly between them. To move to a new location, he uses Dijkstra's shortest path algorithm (Cormen, Leiserson, & Rivest 1989) to identify a collision-free path. In contrast, WhizLow's navigation planner first invokes the A* algorithm to determine an approximate collision-free path on a 2D representation of the 3D world's terrain. However, this only represents an approximate path because it is found by searching through a discretized representation of the terrain. It is critical that control points, i.e., the coordinates determining the actual path to be navigated, be interpolated in a manner that (1) enables the agent's movement to appear smooth and continuous and (2) guarantees retaining the collision-free property. To achieve this natural behavior, the navigation planner generates a Bezier spline that interpolates the discretized path from the avatar's current location, through each successive control point, to the target destination.

To affect their environment, pedagogical agents need a repertoire of motor actions. These generally fall into three categories: speech, control of the agent's body, and control of the learning environment. Speech is typically generated as a text string to speak to a student or another agent. This string might be displayed as is or sent to a speech synthesizer. Control of the agent's body may involve playing existing animation clips for the whole body or may be decomposed into separate motor commands to control gaze, facial expression, gestures, object manipulations, and locomotion. (This issue is discussed in further detail in Section 4.2.) Finally, the agent may need to control the learning environment. For example, to manipulate an object, Steve sends a message to the virtual reality software to generate the appropriate motions of his body and then sends a separate message to the simulator to cause the desired change (e.g., to push a button). Actions in the environment are not restricted to physical behaviors directly performed by the agent. For example, Herman changes the background music to reflect the student's progress. To contextualize the score, he tracks the state of the task model and sequences the elements of the music so that, as progress is made toward successful completion of subtasks, the number of musical voices added increases.

For modularity, it is useful to insulate an agent’s cognitive capabilities from the details of its motor capabilities. For example, Steve’s cognitive module, which controls his behavior, outputs abstract motor commands such as look at an object, move to an object, point at an object, manipulate an object (in various ways), and speak to someone. A separate motor control module decomposes these into detailed messages sent to the simulator, the virtual reality software, and the speech synthesizer. This layered approach means that Steve’s cognition is independent of the details of these other pieces of software, and even of the details of Steve’s body. Because this architecture makes it easy to plug in different bodies, we can evaluate the tradeoffs among them. Steve uses a similarly layered approach on the perception side, to insulate the cognitive module from the particular types of input devices used.

4.2 Behavioral Building Blocks

Designing the behavior of an agent requires addressing two issues. This section addresses the first issue: designing the building blocks from which the agent’s behavior will be generated. The next section discusses the second issue: developing the code that will select and combine the right building blocks to respond appropriately to the dynamically unfolding tutorial situation.

4.2.1 Behavior Spaces

The behavior space approach is the most common method for generating the behavior of a pedagogical agent. A behavior space is a library of behavior fragments. To generate the behavior of the agent, a behavior sequencing engine dynamically strings these fragments together at runtime. When this is done well, the agent’s behavior appears seamless to the student as it provides visually contextualized problem-solving advice.

Figure 12 illustrates the basic idea. It shows a behavior space with three types of behavior fragments: visual segments serving as the agent’s repertoire of movements (depicted in the figure as a drawing of the character), audio clips serving as the agent’s repertoire of utterances (depicted as an audio wave), and segments of background music (depicted as a musical note). The arrows in the behavior space represent the behavior fragments selected by the behavior sequencing engine for a particular interaction with the student, and the lower section of the figure shows how the engine combines them to generate the agent’s behavior and accompanying music.

Creating the behavior fragments for a behavior space can range from simple to quite complex depending on the desired quality of animation. Musical segments are simply audio clips of different varieties of music to create different moods, and utterance segments are typically just voice recordings. A visual segment of the agent could be a simple bitmap image of the agent in a particular pose, a graphical animation sequence of the agent moving from one pose to another, or even an image or video clip of a real person. All three approaches have been used in existing pedagogical agents.

To allow the behavior sequencing engine to select appropriate behavior fragments at runtime, each fragment must be associated with additional information describing its content. For example, behavior fragments in the behavior space for Herman the Bug are indexed ontologically, intentionally, and rhetorically. An ontological index is imposed on explanatory behaviors. Each behavior is labeled with the structure and function of the aspects of the primary pedagogical object that the agent discusses in that segment. For example,

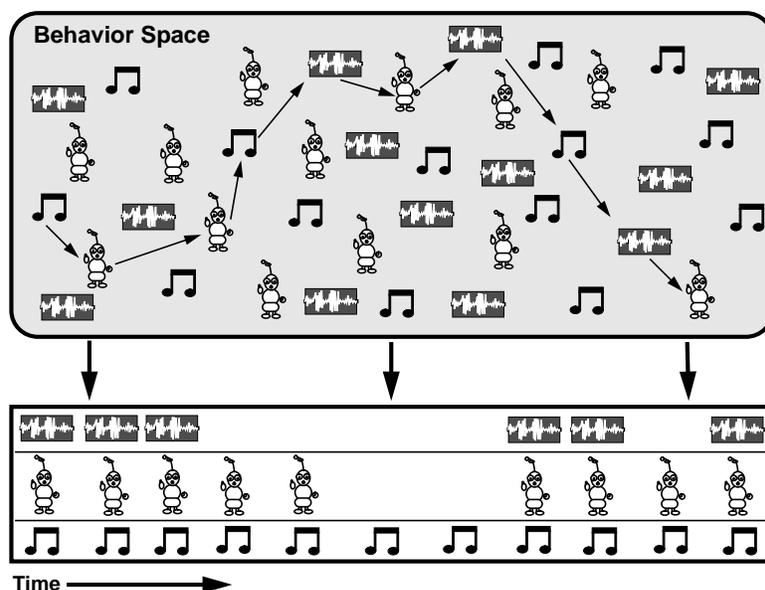


Figure 12: Behavior Space

explanatory segments in Herman’s behavior space are labeled by (1) the type of botanical structures discussed, e.g., anatomical structures such as roots, stems, and leaves, and by (2) the physiological functions they perform, e.g., photosynthesis. An intentional index is imposed on advisory behaviors. Given a problem-solving goal, intentional indices enable the sequencing engine to identify the advisory behaviors that help the student achieve the goal. For example, one of Herman’s behaviors indicates that it should be exhibited when a student is experiencing difficulty with a “low water table” environment. Finally, a rhetorical index is imposed on audio segments. This indicates the rhetorical role played by each clip, e.g., an introductory remark or interjection.

The following example of behavior sequencing in Herman the Bug illustrates this process. If Herman intervenes in a lesson, say because the student is unable to decide on a leaf type, the behavior sequencing engine first selects a topic to provide advice about, some component of the plant being constructed. The engine then chooses how direct a hint to provide: an indirect hint may talk about the functional constraints that a choice must satisfy, whereas a direct hint proposes a specific choice. The level of directness then helps to determine the types of media to be used in the presentation: indirect hints tend to be realized as animated depictions of the relationships between environmental factors and the plant components, while direct hints are usually rendered as speech. Finally, a suitable coherent set of media elements with the selected media characteristics are chosen and sequenced.

One of the biggest challenges in designing a behavior space and a sequencing engine is ensuring visual coherence of the agent’s behavior at runtime. When done poorly, the agent’s behavior will appear discontinuous at the seams of the behavior fragments. For some pedagogical purposes, this may not be serious, but it will certainly detract from the believability of the agent, and it may be distracting to the student. Thus, to assist the sequencing engine in assembling behaviors that exhibit visual coherence, it is critical that the specifications for the animated segments take into account continuity. One simple technique employed by some behavior sequencing engines is the use of visual bookending. Visually bookended animations begin and end with frames that are identical. Just as walk

cycles and looped backgrounds can be seamlessly composed, visually bookended animated behaviors can be joined in any order and the global behavior will always be flawlessly continuous. Although it is impractical for all visual segments to begin and end with the same frame, judicious use of this technique can greatly simplify the sequencing engine’s job.

More generally, the design of behavior spaces can exploit lessons and methods from the film industry. Because the birth and maturation of the film medium over the past century has precipitated the development of a visual language with its own syntax and semantics (Monaco 1981), the “grammar” of this language can be employed in all aspects of the agent’s behaviors. Careful selection of the agent’s behaviors, its accouterments (e.g., props such as microscopes, jetpacks, etc.), and visual expressions of its emotive state (Bates 1994) can emphasize the most salient aspects of the domain for the current problem-solving context.

Many animated agents employ variants of the behavior space approach. Vincent (Paiva & Machado 1998), an animated pedagogical agent for on-the-job training, uses a very simple behavior space, consisting of 4 animation sequences (happy, friendly, sad, and impatient) and 80 utterances. Adele’s animation is produced from a set of bitmap images of her in different poses, which were created from an artist’s drawings. Herman’s behavior sequencing engine orchestrates his actions by selecting and assembling behaviors from a behavior space of 30 animations and 160 audio clips. The animations were rendered by a team of graphic artists and animators. Herman’s engine also employs a large library of runtime-mixable soundtrack elements to dynamically compose a score that complements the agent’s activities.

The PPP Persona and Cosmo also use the behavior space approach. However, to achieve more flexibility in their behavior, they use independent behavior fragments for different visual components of the agent, and the behavior sequencing engine must combine these at runtime. Like Adele, the PPP Persona’s behavior is generated from bitmaps of the agent in different poses. However, the PPP Persona can also use a dynamically generated pointer to refer to specific entities in the world as it provides advice; the sequencing engine must combine an image of the agent in a pointing pose with a pointer drawn from the agent’s hand to the referenced entity.

Cosmo takes this approach much farther. Depending on the physical and pedagogical contexts in which Cosmo will deliver advice, at runtime each “frame” (at a rate of approximately 15/second) is assembled from independent components for torsos, heads, and arms. Dynamic head assembly provides flexibility in gaze direction, while dynamic arm assembly provides flexibility in performing deictic and emotive gestures. Finally, Cosmo exhibits vocal flexibility by dynamically splicing in referring expression phrases to voice clip sequences. For example, this technique enables Cosmo to take into account the physical and dialogue contexts to alternatively refer to an object or group of objects with a proximal demonstrative (“this”), a non-proximal demonstrative (“those”), or perhaps with pronominalization (“it”). Although it is more difficult to dynamically combine body fragments at runtime, the different possible combinations allow for a wider repertoire of behaviors. Cosmo still follows the behavior space approach, since he relies on behavior fragments created ahead of time by designers, but the granularity of his fragments is clearly smaller than an agent like Herman.

The behavior space approach to behavior generation offers an important advantage over the alternate techniques described below: it provides very high quality animations. The granularity of the “building block” is relatively high, and skilled animators have significant control over the process before runtime, so the overall visual impact can at times be quite striking. However, the behavior space suffers from several disadvantages. It is labor inten-

sive (requiring much development time by the animation staff), and because it involves 2D graphics, the student's viewpoint is fixed. Perhaps most lacking, however, is the degree of flexibility that can be exhibited by these agents. Because it is not a fundamentally generative approach, designers must anticipate all of the behavior fragments and develop robust rules for assembling them together.

4.2.2 Generating Behavior Dynamically

To achieve more flexibility, the alternative approach is to completely generate behavior as it is needed, without reusing any canned animation segments or even individual frames. This approach has been used for several systems described in Section 3, including Jack (Badler, Phillips, & Webber 1993), Steve, and WhizLow. These characters each include a 3D graphical model of the agent, segmented into its movable parts. In addition, each includes algorithms that can take a specification of a desired posture and generate the appropriate body motions to transition from the agent's current posture to the desired one. For example, given an object that Steve should look at, an algorithm generates an animation path for his head to follow. The difficulty lies in the fact that typically a number of body parts must move in concert. For instance, even in the simple gaze example, Steve may have to turn his eyes, head, neck, shoulders, and torso, all subject to constraints on their flexibility, and these must move at differential speeds to look natural.

This generative approach works for speech as well as animation. While the behavior space approach pieces together pre-recorded voice clips, the text-to-speech synthesizers used by Steve, Adele, and WhizLow generate speech from individual phonemes. These synthesizers can also apply a wide variety of prosodic transformations. For example, the synthesizer could be instructed to speak an utterance in an angry tone of voice or a more polite tone depending on the context. A wide variety of commercial and public domain speech synthesizers with such capabilities are currently available.

The flexibility of this generative approach to animation and speech comes at a price: it is difficult to achieve the same level of quality that is possible within a handcrafted animation or speech fragment. For now, the designer of a new application must weigh the tradeoff between flexibility and quality. Further research on computer animation and speech synthesis is likely to decrease the difference in quality between the two approaches, making the generative approach increasingly attractive.

4.2.3 Tools for Creating Behavioral Building Blocks

Most of the projects described in this paper have involved people with graphics and animation expertise to create the behavioral building blocks. For projects without this luxury, tools are rapidly becoming available to allow designers to add animated characters to their learning environment even in the absence of such expertise. For example, Microsoft Agent² and Adele's animated persona³ are both available free for download over the World Wide Web. Both provide animated characters with some existing behaviors as well as the ability to create new characters and add new behaviors. Both employ the behavior space approach for animation while using speech synthesizers for voice. In contrast, Jack⁴, available as a commercial product, supports dynamically generated behavior. The increasing availability

²<http://msdn.microsoft.com/workshop/imedia/agent/default.asp>

³<http://www.isi.edu/isd/carte/carte-demos.htm>

⁴<http://www.transom.com/>

of tools for creating animated characters will greatly simplify the development of animated pedagogical agents.

However, creating the behavioral building blocks for an animated character is only the first challenge in developing an animated pedagogical agent. The next challenge is developing the code that will select and combine the right building blocks to respond appropriately to the dynamically unfolding tutorial situation. We now turn to that issue.

4.3 Behavior Control

Controlling the behavior of an animated pedagogical agent requires attention to many issues. Like any other autonomous agent, the agent must be able to react to a dynamic environment. Additionally, like any intelligent tutoring system or learning companion, the agent must carry on a coherent dialogue with the student, and it must make pedagogical decisions, such as when to intervene and what types of information to provide. On top of these considerations, an animated agent must additionally provide appropriate control of its body, complementing its verbal utterances with appropriate nonverbal behavior. The presence of a body marks a significant shift in the problem of behavior control; while a typical tutoring system's behavior is relatively discrete, providing occasional, atomic interventions, nonverbal behavior necessitates more continuous control. In this section we focus on these additional control problems raised by animated agents, and we survey current approaches.

The key to maintaining coherent behavior in the face of a dynamic environment is to maintain a rich representation of context. The ability to react to unexpected events and handle interruptions is crucial for pedagogical agents, yet it threatens the overall coherence of the agent's behavior. A good representation of context allows the agent to be responsive while maintaining its overall focus. Animated pedagogical agents must maintain at least the following three types of context.

Pedagogical context The pedagogical context includes the instructional goals and a model of the student's knowledge. This area has been studied extensively by past researchers; work in animated pedagogical agents to date has contributed relatively little on this issue.

Task context The task context represents the state of the student's and agent's problem solving. This includes the goals of the task, the current state of the learning environment, and the actions that will be needed to complete the task. For example, Steve and Adele model tasks using a hierarchical partial-order plan representation, which they generate automatically using task decomposition planning (Sacerdoti 1977). As the task proceeds, they continually monitor the state of the virtual world, and they use the task model to maintain a plan for how to complete the task, using a variant of partial-order planning techniques (Weld 1994). Because Herman the Bug provides problem-solving advice for design-centered problem solving, he maintains a task model that includes knowledge about the active design constraints, the subtask the student is currently addressing, and a history of her design decisions. Cosmo maintains analogous knowledge about each of the factors bearing on students' problem-solving decisions in the Internet Protocol Advisor learning environment.

Dialogue context The dialogue context represents the state of the collaborative interaction between the student and the agent. This may include many types of information: a focus stack (Grosz & Sidner 1986) representing the hierarchy of tasks, subtasks,

and actions in which the agent and student are currently engaged; the state of their interaction on the current task step (for instance, the state might be that the agent has explained what must be done next but neither he nor the student has done it); a record of whether the agent or student is currently responsible for completing the task (this task initiative can change during a mixed-initiative interaction); the last answer the agent gave, in case the student asks a follow-up question; and the actions that the agent and student have already taken. While this list is not exhaustive, it captures the most important items used in current animated pedagogical agents.

Given a rich representation of context, much of an agent's nonverbal behavior can be generated dynamically in response to the current situation. In Steve, nonverbal behavior is generated in several different layers. Some elements are generated as deliberate acts in Steve's cognitive module. This includes such things as looking at someone when waiting for them or listening to them or releasing the conversational turn, nodding the head when Steve is informed of something or when the student takes an appropriate action, and shaking the head when the student makes a mistake. Other actions are generated in his motor control module to accompany motor commands from the cognitive module. For example, Steve looks where he is going, looks at an object immediately before manipulating or pointing at it, looks at someone immediately before speaking to them, and changes facial expression to a "speaking face" (i.e., mouth open and eyebrows slightly raised) when speaking. Finally, low-level behavior that requires a frame-by-frame update is implemented in the virtual reality software. This includes the animation of Steve's locomotion and arm movements, periodic blinking, slight periodic movement of the lips when speaking, and tracking abilities of Steve's gaze.

The approach to behavior generation discussed so far can be viewed as a mapping from a representation of context to the next appropriate behavioral action. The resulting behavior is coherent to the extent that the regularities of human conversation are built into the mapping. This approach is similar to the schemata approach to explanation generation pioneered by McKeown (McKeown 1985). The other common approach to explanation generation is to plan a coherent sequence of utterances by searching through alternative sequences until one is found that satisfies all coherence constraints (Hovy 1993; Moore 1995). This approach has been adapted to the problem of generating the behavior of an animated agent by André et al. (André & Rist 1996; André, Rist, & Müller 1999) and implemented in their PPP Persona.

Their approach is illustrated in Figure 13. The planning process starts with an abstract communicative goal (e.g., **provide-information** in the figure). The planner's presentation knowledge is in the form of goal decomposition methods called "presentation strategies." In the figure, each nonterminal node in the tree represents a communicative goal, and its children represent one possible presentation strategy for achieving it. For example, the goal **provide-information** can be achieved by **introduce** followed by a sequence of **elaborate** acts. Each presentation strategy captures a rhetorical structure found in human discourse, based largely on Rhetorical Structure Theory (Mann & Thompson 1987), and each has applicability conditions that specify when the strategy may be used and constrain the variables to be instantiated. Given the top-level communicative goal, the presentation planner tries to find a matching presentation strategy, and it posts the inferior acts of this strategy as new subgoals. If a subgoal cannot be achieved, the presentation planner backtracks and tries another strategy. The process is repeated until all leaves of the tree are elementary presentation acts. (A variant of the PPP Persona called WebPersona allows

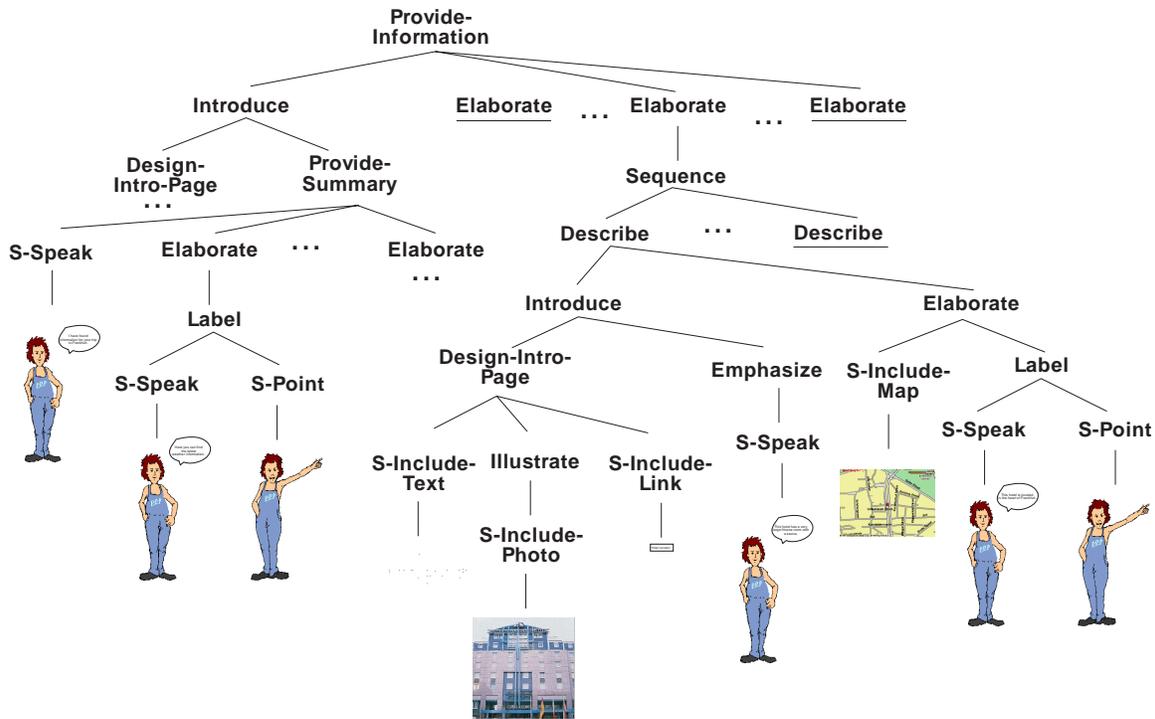


Figure 13: A Presentation Plan

some other types of leaves as well.) Thus, the leaves of the tree in Figure 13 represent the planned presentation, and the tree represents its rhetorical structure.

This presentation script is forwarded to a Persona Engine, which executes it by dynamically merging it with low-level navigation acts (when the agent has to move to a new position on the screen), idle-time acts (to give the agent lifelike behavior when idle), and reactive behaviors (so that the agent can react to user interactions). The Persona Engine decomposes the persona behaviors at the leaves of the presentation plan into more primitive animation sequences and combines these with unplanned behaviors such as idle-time actions (breathing or tapping a foot) and reactive behaviors (such as hanging suspended when the user picks up and moves the persona with the mouse). When behavior execution begins, the persona follows the schedule in the presentation plan. However, since the Persona Engine may execute additional actions, this in turn may require the schedule to be updated, subject to the constraints of the presentation plan. The result is behavior that is adaptive and interruptible, while maintaining coherence to the extent possible.

One of the most difficult yet important issues in controlling the behavior of an animated agent is the timing of its nonverbal actions and their synchronization with verbal utterances. Relatively small changes in timing or synchronization can significantly change people's interpretation or their judgement of the agent. André et al. (André, Rist, & Müller 1999) address timing issues through explicit temporal reasoning. Each presentation strategy includes a set of temporal constraints over its inferior acts. Constraints may include Allen's qualitative temporal relations (Allen 1983) relating pairs of acts, as well as quantitative inequality constraints on the start and end times of the acts. Any presentation whose temporal constraints become inconsistent during planning is eliminated from further consideration.

One important area for further research is the synchronization of nonverbal acts with speech at the level of individual words or syllables. This capability is needed to support many features of human conversation, such as the use of gestures, head nods, and eyebrow movements to highlight emphasized words. Most current animated agents are incapable of such precise timing. One exception is the work of Cassell and her colleagues (Cassell *et al.* 1994a). However, they achieve their synchronization through a multi-pass algorithm that generates an animation file for two synthetic, conversational agents. Achieving a similar degree of synchronization during a real-time dialogue with a human student is a more challenging problem that will require further research.

4.4 Believability

Because of the immediate and deep affinity that people seem to develop for these interactive lifelike characters, the direct pedagogical benefits that pedagogical agents provide are perhaps exceeded by their motivational benefits. By creating the illusion of life, dynamically animated agents have the potential to significantly increase the time that people seek to spend with educational software, and recent advances in affordable graphics hardware are beginning to make the widespread distribution of real-time animation technology a reality. Endowing animated agents with believable, lifelike qualities has been the subject of much recent research (Bates 1994; Tu & Terzopoulos 1994; Granieri *et al.* 1995; Blumberg & Galyean 1995; Kurlander & Ling 1995; Maes *et al.* 1995).

Believability is a product of two forces: (1) the visual qualities of the agent and (2) the computational properties of the behavior control system that creates its behaviors in response to evolving interactions with the user. The behavior canon of the animated film (Noake 1988; Jones 1989; Lenburg 1993) has much to say about aesthetics, movement, and character development, and the pedagogical goals of learning environments impose additional requirements on character behaviors. In particular, techniques for increasing the believability of animated pedagogical agents should satisfy the following criteria:

Situated Liveness Throughout problem-solving sessions, agents should remain “alive” by continuing to exhibit behaviors that indicate their awareness of events playing out in the learning environment, e.g., they can visually track students’ activities and provide anticipatory cues (Thomas & Johnston 1981) to signal their upcoming actions.

Controlled Visual Impact Some behaviors such as moving from one location to another have high visual impact, while others, such as small head movements, have low visual impact. In general, the higher the visual impact, the more interesting a behavior will be, but agents must control the visual impact of their behaviors in such a manner that they do not divert students’ attention at critical junctures.

Complex Behavior Patterns Because students will interact with animated pedagogical agents over extended periods of time, it is critical that agents’ behavior patterns be sufficiently complex that they cannot be quickly induced. Easily recognized behavior patterns significantly reduce believability.

Natural Unobtrusive Behavior It is critical that students’ attention not be drawn to agents because they behave unnaturally. For example, a common problem in early implementations of any pedagogical agent is that the designer has neglected to have them assume a reasonable stance or blink. Omissions such as these typically result in surprisingly odd behaviors.

Achieving believability in animated pedagogical agents poses three major challenges. First, the primary goal of pedagogical agents is to promote learning, and any agent behaviors that would interfere with students' problem-solving—no matter how much these behaviors might contribute to believability—would be inappropriate. For example, if the agent were to cartwheel across the screen when the student was grappling with a difficult problem, the student's concentration would be immediately broken. Second, believability-enhancing behaviors must complement (and somehow be dynamically interleaved with) the advisory and explanatory behaviors that pedagogical agents perform. Third, if observers see that an agent is acting like a simple automaton, believability is either substantially diminished or eliminated altogether.

To achieve believability, agents typically exhibit a variety of believability-enhancing behaviors that are in addition to advisory and "attending" behaviors. For example, the PPP Persona exhibits "idle-time" behaviors such as breathing and foot-tapping to achieve believability. To deal with the concerns of controlled visual impact for sensitive pedagogical situations in which the student must focus his attention on problem-solving, a competition-based believability-enhancing technique is used by one version of the Herman agent. At each moment, the strongest eligible behavior is heuristically selected as the winner and is exhibited. The algorithm takes into account the probable visual impact of candidate behaviors so that behaviors inhabiting upper strata of the "impact spectrum" are rewarded when the student is addressing less critical sub-problems.

Throughout learning sessions, the agent attends to students' problem-solving activities. Believability-enhancing behaviors compete with one another for the right to be exhibited. When the agent is not giving advice, he is kept "alive" by a sequencing engine that enables it to perform a large repertoire of contextually appropriate, believability-enhancing behaviors such as visual focusing (e.g., motion-attracted head movements), re-orientation (e.g., standing up, lying down), locomotion (e.g., walking across the scene), body movements (e.g., back scratching, head scratching), restlessness (e.g., toe tapping, body shifting), and prop-based movements (e.g., glasses cleaning). When a student is solving an unimportant sub-problem, Herman is more likely to perform an interesting prop-based behavior such as cleaning his glasses or a locomotive behavior such as jumping across the screen. The net result of the ongoing competition is that the agent behaves in a manner that significantly increases its believability without sacrificing pedagogical effectiveness.

4.5 Emotion

Engaging, lifelike pedagogical agents that are visually expressive could clearly communicate problem-solving advice and simultaneously have a strong motivating effect on learners. If they could draw on a rich repertoire of emotive behaviors to exhibit contextually appropriate facial expressions and expressive gestures, they could exploit the visual channel to advise, encourage, and empathize with learners. However, enabling lifelike pedagogical agents to communicate the *affective* content of problem-solving advice poses serious challenges. Agents' full-body emotive behaviors must support expressive movements and visually complement the problem-solving advice they deliver. Moreover, these behaviors must be planned and coordinated in real time in response to learners' progress. In short, to create the illusion of life typified by well crafted animated characters, animated pedagogical agents must be able to communicate through both visual and aural channels.

To be maximally entertaining, animated characters must be able to express many different kinds of emotion. As different social situations arise, they must be able to convey

emotions such as happiness, elation, sadness, fear, envy, shame, and gloating. In a similar fashion, because lifelike pedagogical agents should be able to communicate with a broad range of speech acts, they should be able to visually support these speech acts with an equally broad range of emotive behaviors. However, because their role is primarily to facilitate positive learning experiences, only a critical subset of the full range of emotive expression is essential for pedagogical agents. For example, they should be able to exhibit body language that expresses joy and excitement when learners do well, inquisitiveness for uncertain situations (such as when rhetorical questions are posed), and disappointment when problem-solving progress is less than optimal. The Cosmo agent, for instance, can scratch his head in wonderment when he poses a rhetorical question.

Cosmo illustrates how an animated pedagogical agent using the behavior space approach can employ contextually appropriate emotive behaviors. Cosmo employs an emotive-kinesthetic behavior sequencing framework for dynamically sequencing his full-body emotive expressions. Creating an animated pedagogical agent with this framework consists of three phases, each of which is a special case of the phases in the general behavior space approach described above. First, designers add behavior fragments representing emotive behavior to the behavior space. For example, Cosmo includes emotive behavior fragments for his facial expressions (with eyes, eyebrows, and mouth) and gestures (with arms and hands). Second, these behavior fragments must be indexed by their emotional intent (i.e., which emotion is exhibited) and their kinesthetic expression (i.e., how it is exhibited). Third, the behavior sequencing engine must integrate the emotive behavior fragments into the agent's behavior in appropriate situations. For example, Cosmo's emotive-kinesthetic behavior sequencing engine dynamically plans full-body emotive behaviors in real time by selecting relevant pedagogical speech acts and then assembling appropriate visual behaviors. By associating appropriate emotive behaviors with different pedagogical speech act categories (e.g., empathy when providing negative feedback), it can weave small expressive behaviors into larger visually continuous ones that are then exhibited by the agent in response to learners' problem-solving activities.

Both emotive behavior sequencing and its counterpart, affective student modeling, in which users' emotive state is tracked (Picard 1997), will play important roles in future pedagogical agent research. There is currently considerable research activity on computational models of emotion, and a variety of useful frameworks are now available. Research on applying such models to interactive learning environments, on the other hand, has only begun (Elliott, Rickel, & Lester 1999).

4.6 Platform and Networking Issues

All successful animated pedagogical agent designs must take into account the capabilities of the platform and network that are intended to be used. At the present time, high-fidelity interactive agents with dynamically generated behavior can only run on configurations with high processor speed, powerful graphics acceleration, and low latency. For applications where such power is not guaranteed to be available, compromises must be made. The behavior space approach can be used in place of the dynamic behavior generation approach in order to reduce real-time rendering requirements. Reducing the repertoire of gestures can also reduce processing requirements. For example, the "Verbots" created by Virtual Personalities, Inc.⁵ have limited gestures other than lip movement; these agents can run on most Pentium personal computers without graphics acceleration.

⁵<http://www.vperson.com>

The problem of integrating pedagogical agents into Web-based learning materials is an interesting case in point. The Web has become the delivery mechanism of choice for on-line courses. At the same time, Web-based instruction can be very impersonal, with limited ability to adapt and respond to the user. An agent that can integrate with Web-based materials is desirable both because it can be applied to a range of course materials and because it can improve the interactivity and responsiveness of such materials.

The most difficult technical problem associated with Web-based agents is reconciling the highly interactive nature of face-to-face interaction with the slow response times of the Web. In typical Web-based courseware delivery systems, the student must choose a response, submit it to a remote server, and wait for the server to send back a new page. Animated pedagogical agents, on the other hand, need to be able to respond to a continuous stream of student actions, watching what the student is doing, nodding in agreement, interrupting if the student is performing an inappropriate action, and responding to student interruptions. It is difficult to achieve such interactivity if every action must be routed through a central HTTP server.

Two Web-based architectures for animated pedagogical agents, the PPP Persona and Adele, both address this problem by moving reactive agent behavior from the server to the client. The PPP Persona compiles the agent behavior into an efficient state machine that is then downloaded to the client for execution. The presentation planning capability, on the other hand, resides on the central server. In the case of Adele, a solution plan for the given case or problem is downloaded, and is executed by a lightweight student monitoring engine. This approach requires a more sophisticated engine to run on the client side, capable of a range of different types of pedagogical interactions. Nevertheless, the engine remains simple enough to execute on a client computer with a reasonable amount of memory and processor speed. Focusing on one case or problem at a time ensures that the knowledge base employed by the agent at any one time remains small.

The latencies involved in Web-based interaction also become significant when one attempts to coordinate the activities of multiple students on different computers. Adele must address this problem when students work together on the same case at the same time. Separate copies of Adele run on each client machine. Student events are shared between Adele engines using Java's RMI protocol. Each Adele persona then reacts to student events as soon as they arrive at each client machine. This gives the impression at each station of rapid response, even if events are not occurring simultaneously at all client computers.

In summary, integration of animated pedagogical agents into Web-based learning materials inevitably entails developing ways of working around the latencies associated with the HTTP and CGI protocols to some extent. Nevertheless, such agents do take advantage of Web browser environment as appropriate. They point students to relevant Web sites and can respond to browsing actions. Thus, they can be easily integrated into a Web-based curriculum, providing a valuable enhancement.

5 Conclusion

Animated pedagogical agents offer enormous promise for interactive learning environments. Though still in the early stages of development, it is becoming apparent that this new generation of learning technologies will have a significant impact on education and training. By broadening the bandwidth of communication to include many of the modalities of human-human tutoring, pedagogical agents are slowly but surely becoming something

akin to what ITS founders envisioned at the inception of the field. Now, rather than being restricted to textual dialogue on a terminal, pedagogical agents are beginning to perform a variety of tasks in surprisingly lifelike ways. What began as complex but nevertheless small prototype systems have quickly become practical. Some of the systems described here will soon be used in on-line courses; others have been (and continue to be) subject to large-scale empirical studies.

Despite the great strides made in honing the communication skills of animated pedagogical agents, much remains to be done. In many ways, the current state of the art represents the early developmental stages of what promises to be a fundamentally new and interesting species of learning technology. This article has set forth the key functionalities that lifelike agents will need to succeed at face-to-face communication. While the ITS community benefits from the confluence of multidisciplinary research in cognition, learning, pedagogy, and AI, animated pedagogical agents will further require the collaboration of communication theorists, linguists, graphics specialists, and animators. These efforts could well establish a new paradigm in computer-assisted learning, glimpses of which we can already catch on the horizon.

6 Acknowledgments

Support for this work was provided by the Office of Naval Research under grant N00014-95-C-0179 and AASERT grant N00014-97-1-0598; the Air Force Research Laboratory under contract F41624-97-C-5018; a gift from Mitsubishi Electric Research Laboratory; an internal research and development grant from the USC Information Sciences Institute; the National Science Foundation under grants CDA-9720395 (Learning and Intelligent Systems Initiative) and IRI-9701503 (CAREER Award Program); the William S. Kenan Institute for Engineering, Technology and Science; the North Carolina State University IntelliMedia Initiative; an industrial gift from Novell; and equipment donations from Apple and IBM. We are grateful to Brad Mott and the members of CARTE for their valuable comments on an earlier draft.

References

- Aïmeur, E.; Dufort, H.; Leibu, D.; and Frasson, C. 1997. Some justifications for the learning by disturbing strategy. In *Proceedings of the Eighth World Conference on Artificial Intelligence in Education*, 119–126. IOS Press.
- Allen, J. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM* 26(11):832–843.
- André, E., and Rist, T. 1996. Coping with temporal constraints in multimedia presentation planning. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, 142–147. Menlo Park, CA: AAAI Press/MIT Press.
- André, E.; Rist, T.; and Müller, J. 1999. Employing AI methods to control the behavior of animated interface agents. *Applied Artificial Intelligence* 13:415–448.
- André, E., ed. 1997. *Proceedings of the IJCAI Workshop on Animated Interface Agents: Making Them Intelligent*.

- Badler, N. I.; Phillips, C. B.; and Webber, B. L. 1993. *Simulating Humans*. New York: Oxford University Press.
- Ball, G.; Ling, D.; Kurlander, D.; Miller, J.; Pugh, D.; Skelly, T.; Stankosky, A.; Thiel, D.; van Dantzich, M.; and Wax, T. 1997. Lifelike computer characters: the persona project at microsoft. In Bradshaw, J., ed., *Software Agents*. Menlo Park, CA: AAAI/MIT Press.
- Bates, J.; Loyall, A.; and Reilly, W. 1992. Integrating reactivity, goals, and emotion in a broad agent. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, 696–701.
- Bates, J. 1994. The role of emotion in believable agents. *Communications of the ACM* 37(7).
- Blumberg, B., and Galyean, T. 1995. Multi-level direction of autonomous creatures for real-time virtual environments. In *Computer Graphics Proceedings*, 47–54.
- Burton, R. R., and Brown, J. S. 1982. An investigation of computer coaching for informal learning activities. In Sleeman, D., and Brown, J., eds., *Intelligent Tutoring Systems*. Academic Press. 79–98.
- Carbonell, J. R. 1970. AI in CAI: An artificial-intelligence approach to computer-assisted instruction. *IEEE Transactions on Man-Machine Systems* 11(4):190–202.
- Cassell, J., and Thorisson, K. R. 1999. The power of a nod and a glance: Envelope vs. emotional feedback in animated conversational agents. *Applied Artificial Intelligence* 13:519–538.
- Cassell, J.; Pelachaud, C.; Badler, N.; Steedman, M.; Achorn, B.; Becket, T.; Douville, B.; Prevost, S.; and Stone, M. 1994a. Animated conversation: Rule-based generation of facial expression, gesture and spoken intonation for multiple conversational agents. In *Proceedings of ACM SIGGRAPH '94*.
- Cassell, J.; Steedman, M.; Badler, N.; Pelachaud, C.; Stone, M.; Douville, B.; Prevost, S.; and Achorn, B. 1994b. Modeling the interaction between speech and gesture. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Chan, T.-W., and Baskin, A. 1990. Learning companion systems. In Frasson, C., and Gauthier, G., eds., *Intelligent Tutoring Systems: At the Crossroads of Artificial Intelligence and Education*. Ablex.
- Chan, T.-W. 1996. Learning companion systems, social learning systems, and the global social learning club. *Journal of Artificial Intelligence in Education* 7(2):125–159.
- Claassen, W. 1992. Generating referring expressions in a multimodal environment. In Dale, R.; Hovy, E.; Rosner, D.; and Stock, O., eds., *Aspects of Automated Natural Language Generation*. Berlin: Springer-Verlag. 247–62.
- Clancey, W. 1983. The epistemology of a rule-based expert system: A framework for explanation. *Artificial Intelligence* 3(3):215–251.
- Cohn, J. F.; Lien, J. J.-J.; Kanade, T.; Hua, W.; and Zlochower, A. 1998. Beyond prototypic expressions: Discriminating subtle changes in the face. In *Proceedings of the IEEE Workshop on Robot and Human Communication (ROMAN '98)*.
- Cormen, T. H.; Leiserson, C. E.; and Rivest, R. L. 1989. *Introduction to Algorithms*. New York: McGraw-Hill.

- Culhane, S. 1988. *Animation from Script to Screen*. New York: St. Martin's Press.
- Deutsch, B. G. 1974. The structure of task oriented dialogs. In *Proceedings of the IEEE Speech Symposium*. Pittsburgh, PA: Carnegie-Mellon University. Also available as Stanford Research Institute Technical Note 90.
- Dillenbourg, P. 1996. Some technical implications of distributed cognition on the design of interactive learning environments. *Journal of Artificial Intelligence in Education* 7(2):161–179.
- Elliott, C., and Brzezinski, J. 1998. Autonomous agents as synthetic characters. *AI Magazine* 19(2):13–30.
- Elliott, C.; Rickel, J.; and Lester, J. 1999. Lifelike pedagogical agents and affective computing: An exploratory synthesis. In Wooldridge, M., and Veloso, M., eds., *Artificial Intelligence Today*, volume 1600 of *Lecture Notes in Computer Science*. Springer-Verlag. 195–212.
- Elliott, C. 1992. *The Affective Reasoner: A Process Model of Emotions in a Multi-agent System*. Ph.D. Dissertation, Northwestern University.
- Firby, R. J. 1994. Task networks for controlling continuous processes. In *Proceedings of the Second International Conference on AI Planning Systems*.
- Frasson, C.; Mengelle, T.; Aïmeur, E.; and Gouarderes, G. 1996. An actor-based architecture for intelligent tutoring systems. In *Proceedings of the Third International Conference on Intelligent Tutoring Systems (ITS '96)*, number 1086 in *Lecture Notes in Computer Science*, 57–65. Springer.
- Goldstein, I. P. 1976. The computer as coach: An athletic paradigm for intellectual education. Artificial Intelligence Laboratory Memo 389, Massachusetts Institute of Technology, Cambridge, MA.
- Granieri, J. P.; Becket, W.; Reich, B. D.; Crabtree, J.; and Badler, N. I. 1995. Behavioral control for real-time simulated human agents. In *Proceedings of the 1995 Symposium on Interactive 3D Graphics*, 173–180.
- Grosz, B. J., and Sidner, C. L. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics* 12(3):175–204.
- Hayes-Roth, B., and Doyle, P. 1998. Animate characters. *Autonomous Agents and Multi-Agent Systems* 1(2):195–230.
- Hietala, P., and Niemirepo, T. 1998. The competence of learning companion agents. *International Journal of Artificial Intelligence in Education* 9:178–192.
- Hollan, J. D.; Hutchins, E. L.; and Weitzman, L. 1984. Steamer: An interactive inspectable simulation-based training system. *AI Magazine* 5(2):15–27.
- Hovy, E. 1993. Automated discourse generation using discourse structure relations. *Artificial Intelligence* 63:341–385.
- Johnson, W. L., and Rickel, J. 1998. Steve: An animated pedagogical agent for procedural training in virtual environments. *SIGART Bulletin* 8:16–21.
- Johnson, W. L.; Rickel, J.; Stiles, R.; and Munro, A. 1998. Integrating pedagogical agents into virtual environments. *Presence: Teleoperators and Virtual Environments* 7(6):523–546.

- Jones, C. 1989. *Chuck Amuck: The Life and Times of an Animated Cartoonist*. New York: Avon.
- Kurlander, D., and Ling, D. T. 1995. Planning-based control of interface animation. In *Proceedings of CHI '95*, 472–479.
- Laird, J. E.; Newell, A.; and Rosenbloom, P. S. 1987. Soar: An architecture for general intelligence. *Artificial Intelligence* 33(1):1–64.
- Laurel, B. 1990. Interface agents: Metaphors with character. In Laurel, B., ed., *The Art of Human-Computer Interface Design*. New York: Addison-Wesley.
- Lenburg, J. 1993. *The Great Cartoon Directors*. New York: Da Capo Press.
- Lester, J. C.; Converse, S. A.; Kahler, S. E.; Barlow, S. T.; Stone, B. A.; and Bhogal, R. S. 1997a. The persona effect: Affective impact of animated pedagogical agents. In *Proceedings of CHI '97*, 359–366.
- Lester, J. C.; Converse, S. A.; Stone, B. A.; Kahler, S. E.; and Barlow, S. T. 1997b. Animated pedagogical agents and problem-solving effectiveness: A large-scale empirical evaluation. In *Proceedings of the Eighth World Conference on Artificial Intelligence in Education*, 23–30. IOS Press.
- Lester, J. C.; Voerman, J. L.; Towns, S. G.; and Callaway, C. B. 1999a. Deictic believability: Coordinating gesture, locomotion, and speech in lifelike pedagogical agents. *Applied Artificial Intelligence* 13:383–414.
- Lester, J. C.; Zettlemoyer, L. S.; Gregoire, J.; and Bares, W. H. 1999b. Explanatory lifelike avatars: Performing user-designed tasks in 3d learning environments. In *Proceedings of the Third International Conference on Autonomous Agents*.
- Lester, J. C.; Stone, B. A.; and Stelling, G. D. 1999. Lifelike pedagogical agents for mixed-initiative problem solving in constructivist learning environments. *User Modeling and User-Adapted Interaction* 9:1–44.
- Maes, P.; Darrell, T.; Blumberg, B.; and Pentland, A. 1995. The ALIVE system: Full-body interaction with autonomous agents. In *Proceedings of Computer Animation '95*, 11–18. Geneva, Switzerland: IEEE Press.
- Maes, P. 1994. Agents that reduce work and information overload. *Communications of the ACM* 37(7).
- Mann, W. C., and Thompson, S. A. 1987. Rhetorical structure theory: A theory of text organization. In Polanyi, L., ed., *The Structure of Discourse*. Norwood, NJ: Ablex Publishing Corporation. Also available as USC/Information Sciences Institute RS-87-190.
- Marsella, S. C., and Johnson, W. L. 1998. An instructor's assistant for team-training in dynamic multi-agent virtual worlds. In *Proceedings of the Fourth International Conference on Intelligent Tutoring Systems (ITS '98)*, number 1452 in Lecture Notes in Computer Science, 464–473. Springer.
- McKeown, K. R. 1985. *Text Generation*. Cambridge University Press.
- Mittal, V.; Roth, S.; Moore, J. D.; Mattis, J.; and Carenini, G. 1995. Generating explanatory captions for information graphics. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1276–1283.
- Monaco, J. 1981. *How To Read a Film*. New York: Oxford University Press.

- Moore, J. D. 1995. *Participating in Explanatory Dialogues*. Cambridge, MA: MIT Press.
- Müller, J. P. 1996. *The Design of Intelligent Agents: A Layered Approach*. Number 1177 in Lecture Notes in Artificial Intelligence. Springer.
- Munro, A.; Johnson, M.; Surmon, D.; and Wogulis, J. 1993. Attribute-centered simulation authoring for instruction. In *Proceedings of the World Conference on Artificial Intelligence in Education (AI-ED '93)*, 82–89. Association for the Advancement of Computing in Education.
- Murray, W. R. 1997. Knowledge-based guidance in the CAETI center associate. In *Proceedings of the Eighth World Conference on Artificial Intelligence in Education*, 331–339. IOS Press.
- Nagao, K., and Takeuchi, A. 1994. Social interaction: Multimodal conversation with social agents. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, 22–28. Menlo Park, CA: AAAI Press.
- Noake, R. 1988. *Animation Techniques*. London: Chartwell.
- Noma, T., and Badler, N. I. 1997. A virtual human presenter. In *Proceedings of the IJCAI Workshop on Animated Interface Agents: Making Them Intelligent*, 45–51.
- Paiva, A., and Machado, I. 1998. Vincent, an autonomous pedagogical agent for on-the-job training. In *Proceedings of the Fourth International Conference on Intelligent Tutoring Systems (ITS '98)*, 584–593. Springer.
- Pelachaud, C.; Badler, N. I.; and Steedman, M. 1996. Generating facial expressions for speech. *Cognitive Science* 20(1).
- Picard, R. W. 1997. *Affective Computing*. MIT Press.
- Pierrehumbert, J., and Hirschberg, J. 1990. The meaning of intonational contours in the interpretation of discourse. In Cohen, P.; Morgan, J.; and Pollack, M., eds., *Intentions in Communication*. MIT Press. chapter 14, 271–311.
- Reeves, B., and Nass, C. 1998. *The Media Equation: How People Treat Computers, Television and New Media Like Real People and Places*. New York: CSLI.
- Rickel, J., and Johnson, W. L. 1997a. Integrating pedagogical capabilities in a virtual environment agent. In *Proceedings of the First International Conference on Autonomous Agents*. ACM Press.
- Rickel, J., and Johnson, W. L. 1997b. Intelligent tutoring in virtual reality: A preliminary report. In *Proceedings of the Eighth World Conference on Artificial Intelligence in Education*, 294–301. IOS Press.
- Rickel, J., and Johnson, W. L. 1999a. Animated agents for procedural training in virtual reality: Perception, cognition, and motor control. *Applied Artificial Intelligence* 13:343–382.
- Rickel, J., and Johnson, W. L. 1999b. Virtual humans for team training in virtual reality. In *Proceedings of the Ninth International Conference on Artificial Intelligence in Education*. IOS Press.
- Sacerdoti, E. 1977. *A Structure for Plans and Behavior*. New York: Elsevier North-Holland.

- Shaw, E.; Ganeshan, R.; Johnson, W. L.; and Millar, D. 1999. Building a case for agent-assisted learning as a catalyst for curriculum reform in medical education. In *Proceedings of the Ninth International Conference on Artificial Intelligence in Education*. IOS Press.
- Shaw, E.; Johnson, W. L.; and Ganeshan, R. 1999. Pedagogical agents on the web. In *Proceedings of the Third International Conference on Autonomous Agents*.
- Sleeman, D., and Brown, J., eds. 1982. *Intelligent Tutoring Systems*. Academic Press.
- Smith, R. W., and Hipp, D. R. 1994. *Spoken Natural Language Dialog Systems*. Cambridge, Massachusetts: Oxford University Press.
- Stevens, A.; Roberts, B.; and Stead, L. 1983. The use of a sophisticated graphics interface in computer-assisted instruction. *IEEE Computer Graphics and Applications* 3:25–31.
- Stiles, R.; McCarthy, L.; and Pontecorvo, M. 1995. Training studio: A virtual environment for training. In *Workshop on Simulation and Interaction in Virtual Environments (SIVE-95)*. Iowa City, IW: ACM Press.
- Tambe, M. 1997. Towards flexible teamwork. *Journal of Artificial Intelligence Research* 7:83–124.
- Thomas, F., and Johnston, O. 1981. *The Illusion of Life: Disney Animation*. New York: Walt Disney Productions.
- Thorisson, K. R. 1996. *Communicative Humanoids: A Computational Model of Psychosocial Dialogue Skills*. Ph.D. Dissertation, Massachusetts Institute of Technology.
- Towns, S. G.; Callaway, C. B.; and Lester, J. C. 1998. Generating coordinated natural language and 3D animations for complex spatial explanations. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*.
- Tu, X., and Terzopoulos, D. 1994. Artificial fishes: Physics, locomotion, perception, and behavior. In *Computer Graphics Proceedings*, 43–50.
- Walker, J. H.; Sproull, L.; and Subramani, R. 1994. Using a human face in an interface. In *Proceedings of CHI-94*, 85–91.
- Weld, D. S. 1994. An introduction to least commitment planning. *AI Magazine* 15(4):27–61.
- Wenger, E. 1987. *Artificial Intelligence and Tutoring Systems*. Los Altos, CA: Morgan Kaufmann.