Mixed-Initiative Interaction between Pedagogical Agents and Students in Virtual Environments

Jeff Rickel and W. Lewis Johnson
Information Sciences Institute & Computer Science Department
University of Southern California
4676 Admiralty Way, Marina del Rey, CA 90292-6695
rickel, johnson@isi.edu
http://www.isi.edu/isd/VET/vet.html

Abstract

Virtual reality can broaden the types of interaction between students and computer tutors. As in conventional simulation-based training, the computer can watch students practice tasks, responding to questions and offering advice. However, immersive virtual environments also allow the computer tutor to physically inhabit the virtual world with the student. Such a "pedagogical agent" can physically collaborate with the student on tasks and employ the sorts of nonverbal communication used by human tutors. This paper describes Steve, a pedagogical agent for virtual environments that helps students learn procedural tasks. Steve inhabits the virtual world with students, and he collaborates with them on tasks by gracefully shifting between demonstrating the task and providing assistance while the student performs the task. The paper also describes the subtle ways in which such a pedagogical agent can interact with students through nonverbal communication to achieve more human-like collaboration.

1 Introduction

Virtual environment technology can enhance simulation-based training to make it closer to real-life experience. Students, immersed in a 3D computer simulation of a work environment, improve their skills through practice on realistic tasks. Simulation-based training is especially valuable in domains where real-life practice is expensive or hazardous, such as surgery, air combat, or control of a complex manufacturing process. Virtual environments can potentially extend the range of situations that can be adequately simulated, because they are more suitable than previous technologies for providing realistic perceptual stimuli (e.g., visual, auditory, and haptic) (Durlach & Mavor 1995).

Virtual environment technology also enables intelligent tutoring systems to overcome key limitations of the computer coach paradigm. In the coaching paradigm, the computer watches as the student performs tasks, responding to questions and offering advice (Goldstein 1976). In virtual environments, the coach can physically inhabit the virtual world along with the student. This permits a wider variety of interactions between student and coach: they can physically collaborate on tasks, and they can interact and communicate in nonverbal ways that would be impossible with a traditional disembodied coach.

Towards these goals, we are developing a pedagogical agent called Steve (Soar Training Expert for Virtual Environments). Steve inhabits a virtual environment, continually monitoring the state of the environment and periodically manipulating it through virtual motor actions. Steve both instructs and collaborates with students; his objective is to help students learn to perform procedural tasks, such as operating or repairing complex devices. He can demonstrate how to perform tasks, rationalize his actions, and monitor students performing tasks, providing assistance when it is needed. These capabilities allow Steve to provide valuable assistance to students when a human instructor is unavailable. Steve functions as part of a larger Virtual Environments for Training (VET) system being developed jointly by the USC Information Sciences Institute, the USC Behavioral Technology Laboratory, and Lockheed Martin.

This paper focuses on Steve's interaction with students. After briefly discussing Steve's interface to the rest of the VET system in Section 2, we discuss Steve's current interaction abilities in Section 3. Specifically, we describe how Steve uses a standard plan-based representation of tasks, along with an adaptive method of plan execution, to gracefully shift between performing the task himself and providing assistance while a student performs the task. However, our current work has barely touched on the potential for human-computer interaction in a virtual environment; Section 4 describes several ways Steve could exploit this potential. In order to focus on issues of mixed-initiative interaction between Steve and students, this paper provides only an abbreviated description of the technical details underlying Steve and the motivation behind design de-

cisions; for more information, see (Rickel & Johnson 1997).

2 Steve's World

Before discussing Steve's interaction with students, it is helpful to understand the world as Steve sees it. Steve is but one component in the overall VET system. The VET system is built around a distributed architecture, in which each component runs as a separate process, possibly on a separate workstation. The components communicate by sending messages to one another.

There are several types of components. A simulator component (Munro *et al.* 1993) controls the behavior of the virtual world. The simulation is driven by events, such as the passage of time and the actions taken by humans and agents. In response to these events, the simulator updates the state of the world and broadcasts messages describing the state changes, in terms of objects and attributes. In order to view and control the virtual world, each human participant has a Vista component (Stiles, McCarthy, & Pontecorvo 1995). Vista provides a 3D, immersive, graphical rendering of the virtual world, which human participants view through a head-mounted display. Vista also allows humans to interact with the virtual world using devices such as a 3D mouse or data gloves; when Vista detects such interactions, it broadcasts messages to the other components, such as the simulator. Speech generation components permit agents to talk to human participants; each student has a speech generation component that receives text messages broadcast from other components and generates speech. To allow students to talk to agents, we are currently working on speech recognition components as well. Finally, there are agent components, such as Steve.

Steve consists of two modules: the first, implemented in Soar (Laird, Newell, & Rosenbloom 1987; Newell 1990), handles high-level cognitive processing, and the second handles sensorimotor processing. Roughly speaking, the cognitive module repeatedly chooses Steve's next action, and the sensorimotor module provides Steve's perception of the world and carries out the chosen action. To provide perception to the cognitive module, the sensorimotor module continually monitors the messages from other components. These messages describe changes in the virtual world (in terms of objects and their attributes), actions taken by humans (e.g., a mouse click on a particular object), the student's current field of view (i.e., which objects he can see), and the current time (as measured by a real-time clock). At the beginning of every Soar decision cycle (roughly 10 – 100 times per second), the Soar module asks for the current state of the world, along with any important events that occurred since the last such request (e.g., an action taken by the student). After the cognitive module chooses the next appropriate action, it sends that action to the sensorimotor module, which in turn sends the appropriate lower-level messages to the other components in order to implement the action. Steve's repertoire of actions currently includes object manipulation (e.g., pressing buttons), visual attention (e.g., looking at objects), gestures (e.g., pointing at objects), and speech.

We are currently experimenting with two physical realizations for Steve. The simplest is a virtual hand that can manipulate and point at objects. For example, Figures 1, 2, and 3 show Steve pressing a button, grasping an object, and pointing to an object on a shipboard air compressor. We are also developing methods by which Steve can appear as a full human figure, using the Jack software developed at the University of Pennsylvania (Badler, Phillips, & Webber 1993). For example, Figure 4 shows one Steve agent, represented by a human figure, watching (via dynamic gaze control) a demonstration by another Steve agent, represented by a hand. Although a full human figure is more visually obtrusive and more difficult to control, such a representation may be more appropriate for demonstrating some spatial skills, and it allows a wider variety of nonverbal communication between Steve and students.

3 Interaction between Steve and the Student

Steve can interact with students in two ways: he can demonstrate how to perform a task, and he can monitor a student performing a task, providing assistance when requested. Moreover, Steve can shift gracefully between these two modes. The primary difference between demonstration and monitoring is whether Steve or the student has control over choosing and executing task actions.

Regardless of who is in control (i.e., whether Steve is demonstrating or monitoring), Steve must maintain a plan for completing the current task. This is obvious in the case of demonstration, since Steve must select and execute actions to complete the task. In the case of monitoring, Steve must be able to advise the student, when asked, on the next appropriate action, and he must be able to rationalize his advice in terms of how that action helps complete the task.

Steve uses a standard plan representation (Russell & Norvig 1995). First, each plan consists of a set of steps, each of which is either a primitive action (e.g., push a button) or a complex action (i.e., itself a plan). Second, there may be ordering constraints among the

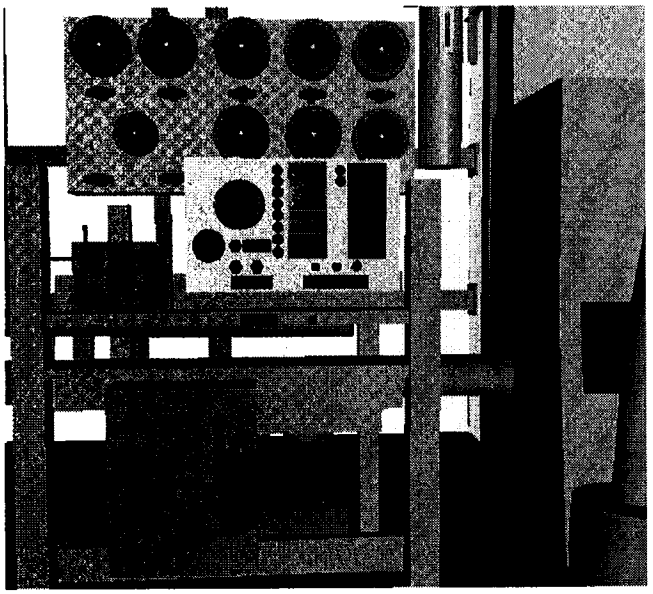

Figure 1: Steve pressing a button

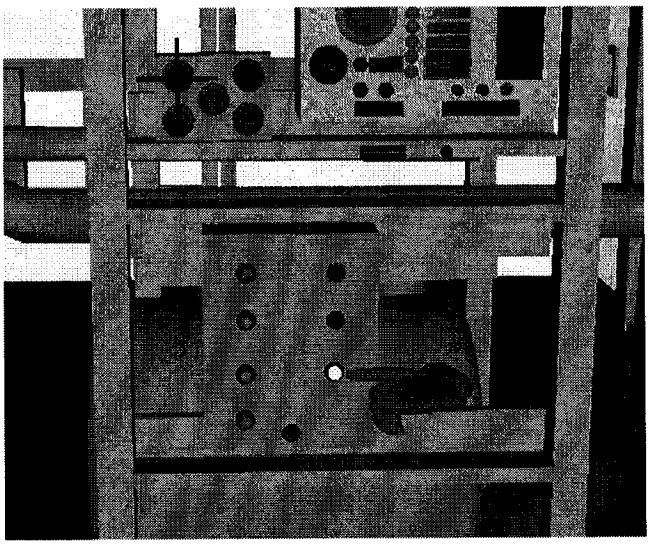

Figure 3: Steve pointing to an indicator light

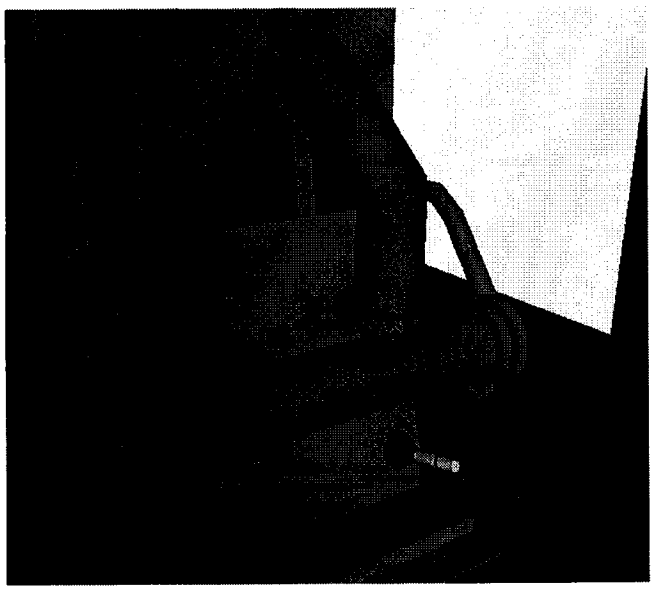

Figure 2: Steve grasping a dipstick

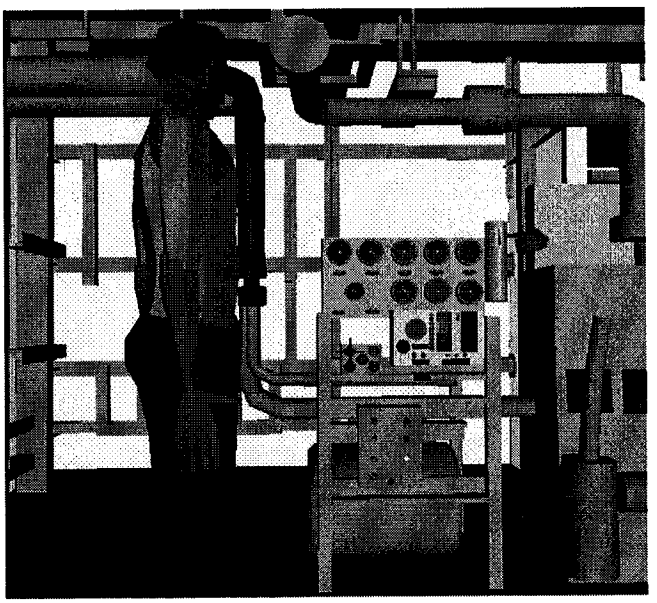

Figure 4: One Steve agent watching another

steps; these constraints define a partial order over the steps. Finally, the role of the steps in the plan is represented by a set of causal links (McAllester & Rosenblitt 1991); each causal link specifies that one step in the plan achieves a goal that is a precondition for another step in the plan (or for termination of the task).

Given a task to demonstrate or monitor, Steve constructs a plan for performing the task, using top-down task decomposition (Sacerdoti 1977). That is, he repeatedly expands any complex step in the evolving plan with the subplan (given to Steve as domain knowledge) for achieving it, until the plan has been fully decomposed. However, Steve cannot simply execute this plan by rote. When monitoring the student, Steve is not in control, so he cannot prevent the student from deviating from the plan. (Many tutoring systems do prevent the student from deviating from the tutor's plan, but our objective is to give the student more freedom.) Even when demonstrating, Steve cannot execute the plan by rote, because the environment itself may be unpredictable. Thus, Steve must be able to adapt the plan to unexpected circumstances. Moreover, he must do so quickly, since he and the student are collaborating on the task in real time.

Thus, Steve continually re-evaluates his plan as the virtual world changes. When he first constructs a plan for the task (as just described), he includes all steps that might be required, even if they are not necessary given the current state of the world. Then, by continually comparing the goals of the plan (both end goals and intermediate ones) against the state of the changing world, Steve repeatedly identifies the subset of the plan that is still relevant to completing the task, using a method analogous to partial-order planning (Weld 1994). As long as some subset of the plan suffices for completing the task, Steve can identify the next appropriate action. This approach is efficient, allowing Steve to interact with the student in real time, yet it still allows Steve to adapt the plan to unexpected events: Steve naturally re-executes parts of the plan that get unexpectedly undone, and he naturally skips over parts of the plan that are unnecessary because their goals were achieved by other means. The key to Steve's ability to adaptively execute plans is his understanding of the causal links in the plan, because they relate the plan's steps to the goals (both end goals and intermediate ones) that those steps depend on (as preconditions) and achieve.

The main difference between demonstration and monitoring is who takes the actions. When demonstrating, Steve performs each action, explaining what he is doing. When monitoring the student, Steve allows the student to take the actions. However, during monitoring, Steve must still perform any sensing actions in the plan (e.g., checking whether a light is on), because sensing actions only change the mental state of the student, which is unobservable. Whenever a sensing action is appropriate, and the object to be sensed is in the student's field of view, Steve takes the action, records the result, and assumes that the student did the same.

Steve and the student can interact in many ways during a task. When Steve is demonstrating the task, the student can interrupt Steve and ask to finish the task, in which case Steve shifts to monitoring the student. When the student is performing the task, the student can ask Steve to demonstrate the next step. He can also ask Steve to recommend the next step. He can ask for the rationale behind Steve's recommendation, and can even ask follow-up questions about Steve's subsequent answers, as shown in Figure 5. Steve can provide the rationale for his recommendations, as well as answer follow-up questions, by combining his knowledge of the causal links in the plan with his knowledge of which parts of the plan are still relevant. Finally, after the task is complete, Steve can answer questions about the rationale behind actions he took during the task. Steve uses Johnson's Debrief system (Johnson 1994) to maintain an episodic memory of the actions he performs and the situations in which he performs them, so he answers such "after-action review" questions by recalling the situation and applying the same question answering ability illustrated in Figure 5. Thus, Steve can collaborate with the student in performing the task as well as explain his knowledge of the task.

4 Future Types of Interaction

As described in the previous section, Steve can shift gracefully from demonstration, where he controls performance of the task, to monitoring, where the student controls performance of the task. However, the decision to shift control is not Steve's. Initially, Steve is asked to either demonstrate or monitor; the request can come from the student or from another tutoring module of the VET system. Subsequently, the student controls the shifts by asking questions, asking to take over the task, or asking for Steve to take over, as described in the last section. Thus, while Steve is capable of shifting control, he is not capable of initiating shifts.

However, previous tutoring systems built by the authors did include methods for automatically shifting control, and these methods could be incorporated into Steve. Like Steve, the TOTS tutor built by Rickel (Rickel 1988) coaches students through procedural tasks. Unlike Steve, TOTS maintains a record of which subtasks the student has been taught. As

```
Steve: I suggest you press the function test button.
Student: Why?
Steve: That action is relevant because we want the drain monitor in
       test mode.
Student: Why?
Steve: That goal is relevant because it will allow us to check the
       alarm lights.
Student: Why?
Steve: That action is relevant because we want to know whether the
       alarm lights are functional.
```

Figure 5: Example explanations generated by Steve

the student performs a task, TOTS continually checks whether the student has sufficient experience to perform the next subtask. If the student has been taught the subtask, perhaps in another context, TOTS relinquishes control to the student; if not, TOTS intervenes to introduce the subtask and demonstrate it to the student. TOTS also intervenes when the student makes a mistake (i.e., performs an inappropriate action). In this case, TOTS describes why the student's action is inappropriate and then demonstrates the appropriate action.

In contrast, the REACT tutor built by Hill and Johnson (Hill & Johnson 1995) only intervenes when it believes the student is at an "impasse". Informally, an impasse is a point at which the student is failing to make progress or has made a mistake that will impede subsequent progress. Operationally, REACT recognizes three types of impasse: (1) the student's action failed (e.g., caused the device he is operating to produce an error message; (2) the student finished all the steps of a subtask without achieving its goals (e.g., one of the steps did not have the intended effect); and (3) the student started a new subtask without completing the previous one. In these cases, REACT explains the cause of the problem to the student and helps him recover. Thus, TOTS and REACT provide criteria by which Steve could decide when to shift control between demonstration and monitoring.

Because Steve, unlike TOTS and REACT, has a physical representation in the virtual world, he could potentially use nonverbal communication to affect control of the task in many subtle ways. He could use gaze or pointing to direct the student's attention to a new danger in the environment. He could use a nod of approval to show agreement with the student's actions, and a nod of disapproval or look of puzzlement to make the student think twice. Cassell et al. (1994) discuss the uses of gaze to coordinate turn-taking in human dialogue, and they describe an implemented program for analogously controlling gaze in a Jack human figure. These subtle methods for influencing the student are much less obtrusive than those used by most current tutoring systems, so they have the potential to influence the direction of the problem-solving task without directly stealing control from the student.

Conversely, while students currently communicate with Steve via a point-and-click interface, a virtual environment opens up other alternatives. Using position sensors on the hands, as well as data gloves, students could use gestures to get Steve's attention (e.g., by raising a hand) and could reference objects in the virtual world by pointing at them. Billinghurst and Savage (1996) developed a virtual environment interface between humans and agents that allows the human to combine speech and gestures, and they report two key advantages of the combination: (1) different types of communication are simpler in one or the other mode, and (2) in cases where either mode alone would be ambiguous, the combination can help disambiguate. Also, Steve could use a student's field of view and hand positions to anticipate errors or recognize confusion. In contrast, a conventional desktop tutor gets little or no information about the student's activity until the student commits to a particular action. The extra bandwidth for human-computer communication in virtual environments works both ways: both the student and computer tutor can influence the direction of their mutual task in more varied, natural ways.

In order to train students on team tasks, we are extending Steve to operate within multi-agent settings. In this context, Steve agents will play roles within the team. Steve's capabilities for performing tasks have been designed with this objective in mind, so we expect relatively few required changes. However, Steve's current tutorial interactions with students must be broadened to include peer-to-peer team interactions. As a

team member, Steve will be less likely to intervene in a student's activities unless the student explicitly asks for help. A request for help will cause Steve to monitor the student more closely and intervene more freely.

Ultimately, we are working toward an agent that can carry on flexible, interactive, tutorial and collaborative dialogues with students. There has been much recent work in this area in the text planning community (e.g., (Cawsey 1992), (Lambert & Carberry 1991), (Lochbaum, Grosz, & Sidner 1990), (Moore 1995), (Walker & Whittaker 1990)), and we expect to build on that work. However, our focus includes not only language but also the nonverbal types of communication facilitated by having a virtual environment where both students and agents are physically situated. Despite our progress, we have only begun to exploit the potential of virtual environments.

5 Acknowledgments

The VET project is funded by the Office of Naval Research, grant N00014-95-C-0179. We gratefully acknowledge the efforts of our colleagues on the project: Randy Stiles, Laurie McCarthy, and Sandeep Tewari at Lockheed Martin, Allen Munro, Mark Johnson, Quentin Pizzini, and Jim Wogulis at the Behavioral Technology Laboratory, and Richard Angros and Erin Shaw at the USC Information Sciences Institute.

References

Badler, N. I.; Phillips, C. B.; and Webber, B. L. 1993. *Simulating Humans.* New York: Oxford University Press.

Billinghurst, M., and Savage, J. 1996. Adding intelligence to the interface. In *Proceedings of the IEEE Virtual Reality Annual International Symposium (VRAIS '96)*, 168–175. Los Alamitos, CA: IEEE Computer Society Press.

Cassell, J.; Pelachaud, C.; Badler, N.; Steedman, M.; Achorn, B.; Becket, T.; Douville, B.; Prevost, S.; and Stone, M. 1994. Animated conversation: Rule-based generation of facial expression, gesture and spoken intonation for multiple conversational agents. In *Proceedings of ACM SIGGRAPH '94.*

Cawsey, A. 1992. *Explanation and Interaction: The Computer Generation of Explanatory Dialogues.* Cambridge, MA: MIT Press.

Durlach, N. I., and Mavor, A. S., eds. 1995. *Virtual Reality: Scientific and Technological Challenges.* Washington, D.C.: National Academy Press.

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.

Hill, Jr., R. W., and Johnson, W. L. 1995. Situated plan attribution. *Journal of Artificial Intelligence in Education* 6(1):35–66.

Johnson, W. L. 1994. Agents that learn to explain themselves. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, 1257–1263. Menlo Park, CA: AAAI Press.

Laird, J. E.; Newell, A.; and Rosenbloom, P. S. 1987. Soar: An architecture for general intelligence. *Artificial Intelligence* 33(1):1–64.

Lambert, L., and Carberry, S. 1991. A tripartite plan-based model of dialogue. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, 47–54. Morgan Kaufmann Publishers.

Lochbaum, K. E.; Grosz, B. J.; and Sidner, C. L. 1990. Models of plans to support communication: An initial report. In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, 485–490. Los Altos, CA: Morgan Kaufmann.

McAllester, D., and Rosenblitt, D. 1991. Systematic nonlinear planning. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, 634–639. Menlo Park, CA: AAAI Press.

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

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.

Newell, A. 1990. *Unified Theories of Cognition.* Cambridge, MA: Harvard University Press.

Rickel, J., and Johnson, W. L. 1997. Integrating pedagogical capabilities in a virtual environment agent. In *Proceedings of the First International Conference on Autonomous Agents.*

Rickel, J. 1988. An intelligent tutoring framework for task-oriented domains. In *Proceedings of the International Conference on Intelligent Tutoring Systems.*

Russell, S., and Norvig, P. 1995. *Artificial Intelligence: A Modern Approach.* Englewood Cliffs, NJ: Prentice Hall.

Sacerdoti, E. 1977. *A Structure for Plans and Behavior.* New York: Elsevier North-Holland.

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.

Walker, M., and Whittaker, S. 1990. Mixed initiative in dialogue: An investigation into discourse segmentation. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, 70–78. Morgan Kaufmann Publishers.

Weld, D. S. 1994. An introduction to least commitment planning. *AI Magazine* 15(4):27–61.