Detecting and Correcting Misconceptions with Lifelike Avatars in 3D Learning Environments

Joël P. Grégoire   Luke S. Zettlemoyer   James C. Lester
Multimedia Laboratory
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206
{jgregoi,lszettle,lester}@eos.ncsu.edu

Abstract: Lifelike pedagogical agents offer significant promise for addressing a central problem posed by learning environments: detecting and correcting students' misconceptions. By designing engaging lifelike avatars and introducing them into task-oriented 3D learning environments, we can enable them to serve a dual communication role. They can serve as students' representatives in learning environments and simultaneously provide realtime advice to address their misconceptions. We describe a three-phase *avatar-based misconception correction* framework in which (1) as students navigate their avatars through 3D learning environments, a *misconception detector* tracks their problem-solving activities by inspecting task networks, (2) when they take sub-optimal or incorrect actions, a *misconception classifier* examines misconception trees to identify the most salient misconceptions, and (3) the avatars employ a *misconception corrector* to intervene with tailored advice. The framework has been implemented in a misconception system for the lifelike avatar of CPU CITY, a 3D learning environment testbed for the domain of fundamentals of computer architecture. In CPU CITY, students direct their lifelike avatar to navigate a 3D world consisting of a virtual computer, to transport data and address packets, and to manipulate computational devices. Informal studies with students interacting with CPU CITY suggest that the framework can is an effective tool for addressing students' misconceptions.

1. Introduction

Research on lifelike pedagogical agents has been the subject of increasing attention in the AI & Education community [1,12,14,19,21]. Because of the immediate and deep affinity that children seem to develop for interactive lifelike characters, their potential benefits to learning effectiveness are substantial. By creating the illusion of life, animated agents offer much promise for increasing both the quality of learning experiences as well as the time that children seek to spend with learning environments. Moreover, recent advances in affordable graphics hardware are beginning to make the widespread distribution of realtime 3D animation software a reality.

Lifelike pedagogical agents have significant potential for addressing a central problem posed by learning environments: detecting and correcting misconceptions. From the classic work on student modeling [6,7] plan recognition [9], and plan attribution [13], researchers have endeavored to unobtrusively track learners' problem-solving activities and dialogues and correct their misconceptions [17]. Lifelike pedagogical agents can enable us to recast these problems into a solvable form. By designing task-oriented 3D learning environments, representing detailed knowledge of their 3D models and layout, and introducing lifelike avatars into these worlds, we can craft engaging educational software that effectively addresses learners' misconceptions.

For the past several years our laboratory has been investigating these issues by iteratively designing, implementing, and evaluating avatar-based 3D learning environments. Previously, we have reported results on task-sensitive cinematography planning for 3D learning environments [2], the habitable 3D learning environments framework [3], and performing student-designed tasks [15]. In this paper, we describe a three-phase avatar-based misconception correction framework:

1. **Misconception Detection**: As the learner solves a problem in a 3D learning environment by directing her avatar to navigate through the world and to manipulate objects within it, a misconception detector tracks her problem-solving actions by inspecting a *task network*.
2. **Misconception Classification**: When she takes sub-optimal actions, a *misconception classifier* examines a *misconception tree* to identify the most salient misconception.
3. **Misconception Correction**: Finally, a misconception corrector directs the avatar to address conceptual problems by examining a *curriculum information network* [24], intervening with verbal advice, and providing tailored responses to follow-up questions she poses.

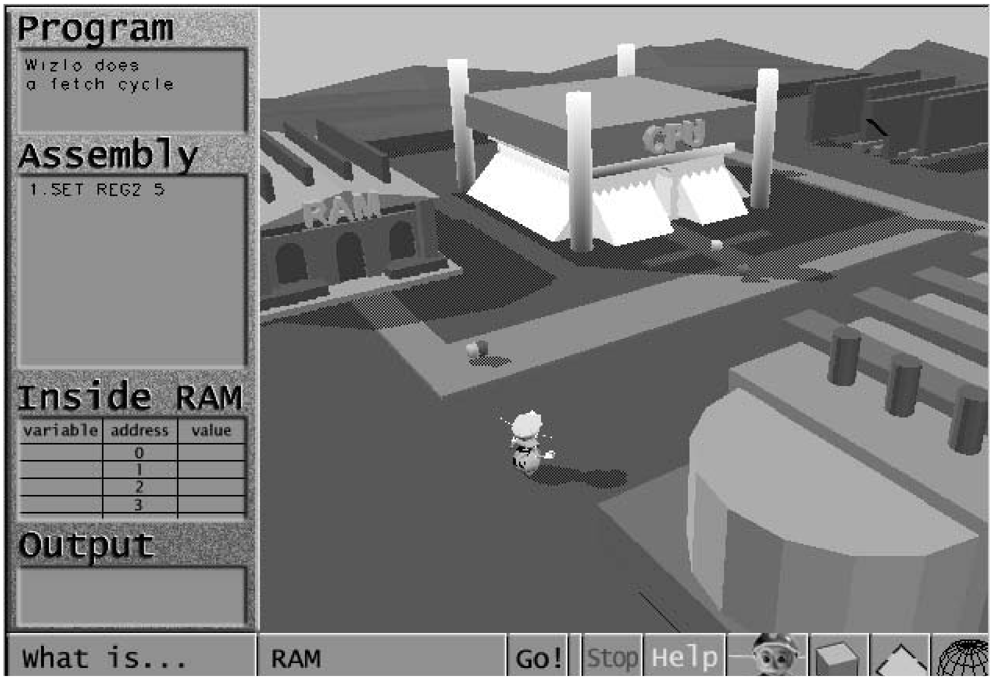
By enabling the avatar to serve in the dual capacity of student representative and advice-giving agent, it tightly couples problem-solving to misconception correction. In short, avatar-based misconception detection and correction provide a critical link between task-oriented conceptual development and addressing learners' misconceptions directly in problem-solving contexts. This framework has been implemented in a lifelike avatar, WHIZLOW, who inhabits the CPU CITY 3D learning environment testbed (Figure 1) for the domain of computer architecture for novices. Given "programming assignments," learners direct WHIZLOW to pick up, transport, and insert data packets into registers to execute their programs. A formative study with students interacting with WHIZLOW suggests that the framework can be an effective tool for addressing misconceptions.

2. Lifelike Avatars in 3D Environments

A particularly promising line of work underway is that of lifelike animated intelligent agents. Because of these agents' compelling visual presence and their high degree of interactivity, there has been a surge of interest in *believable* intelligent characters [1,4,5,8]. As a result of these developments, the AI & Education community is now presented with opportunities for exploring new technologies for lifelike pedagogical agents. Work in this direction is still in its infancy, but progress is being made on two fronts. First, research has begun on a variety of pedagogical agents that can facilitate the construction of component-based tutoring system architectures and communication between their modules [22,23], provide multiple context-sensitive pedagogical strategies [18], reason about multiple agents in learning environments [11], provide assistance to trainers in virtual worlds [16], and act as co-learners [10]. Second, projects have begun to investigate techniques by which animated pedagogical agents can behave in a lifelike manner to communicate effectively with learners both visually and verbally [1,14,19,21]. It is this second category, lifelike animated pedagogical agents, that is the focus of the work described here. In particular, we investigate the marriage of avatar-style problem-solving functionalities with misconception correction functionalities.

Over the course of the past few decades, educators and learning theorists have shifted from an emphasis on rote learning to constructivist learning. The latter emphasizes learning as knowledge construction, which is reflected in the design requirements we impose on avatar-based misconception detection and correction:

- ***Situated problem-solving*:** Because of constructivism's focus on the learner's active role in acquiring new concepts and procedures [20], problem solving should play a central role in learning. For example, to learn procedural tasks, a 3D learning environment could enable students to perform the task directly in the environment. Hence, rather than memorizing an abstract procedure, students should be able to actively solve problems, perhaps by immersively interacting with rich 3D models representing the subject matter.
- ***Non-invasive task monitoring:*** Although continuously querying the student about her moment-to-moment activities would provide up-to-date information about her knowledge and intentions—this would significantly simplify the problems posed by student modeling [6,7], plan recognition [9], and plan attribution [13]—such an invasive technique would distract the learner from her activities and likely result in substantially reduced learning effectiveness.



- ***Embodied advice*:** The traditional approach to introducing explanation facilities into learning environments has been with text-based dialogues. However, given the motivational benefits of lifelike agents [1,12,14], "embodying" explanations in onscreen personae would enable learning environments to provide avatars that were (1) visually present in the world, e.g., immersed in a 3D learning environment, (2) able to exhibit navigational and manipulative behaviors in the world to clearly correct misconceptions, and (3) adept at coordinating their physical behaviors with a running verbal explanation that is tightly coupled to the task the learner was performing.

3. Detecting and Correcting Misconceptions in 3D Environments

The learner's avatar serves two key roles in addressing misconceptions. First, she solves problems by manipulating a joystick to direct her avatar's behaviors. In response to these directives, the avatar performs actions such as picking up objects, carrying them, manipulating devices, and traveling from one location to another. Second, her avatar serves in an advisory capacity by providing explanations. When the student takes actions that indicate she harbors misconceptions about the domain, her avatar intervenes and corrects her misconceptions by providing appropriate advice. Hence, problem solving and advice both play out immersively in the 3D world. All of the avatar's misconception detection/correction functionalities are provided by the architecture shown in Figure 2. To begin, the avatar verbally presents a problem for the student to solve. As the student begins to perform the task to solve the problem, her position and the changes she enacts to the objects she manipulates are reflected in continuous updates made to the *3D world model*, which represents the coordinates and state of all objects in the learning environment. All of her problem-solving activities are tracked by an *avatar misconception handler*, which observes and addresses the misconceptions she exhibits while interacting with the world.

3.1 Misconception Detection

Problem-solving begins when the avatar poses a problem for the learner to solve in the 3D world. To take advantage of the immersive properties of 3D worlds, the avatar-based framework described here focuses on procedural tasks in which learners acquire concepts relating sequences of steps in a coached-practice setting. For example, in the CPU CITY learning environment testbed, students are posed problems about how to perform the fetch-execute cycle in the virtual computer and the avatar coaches their problem-solving activities. Given a particular "programming assignment," their job is to

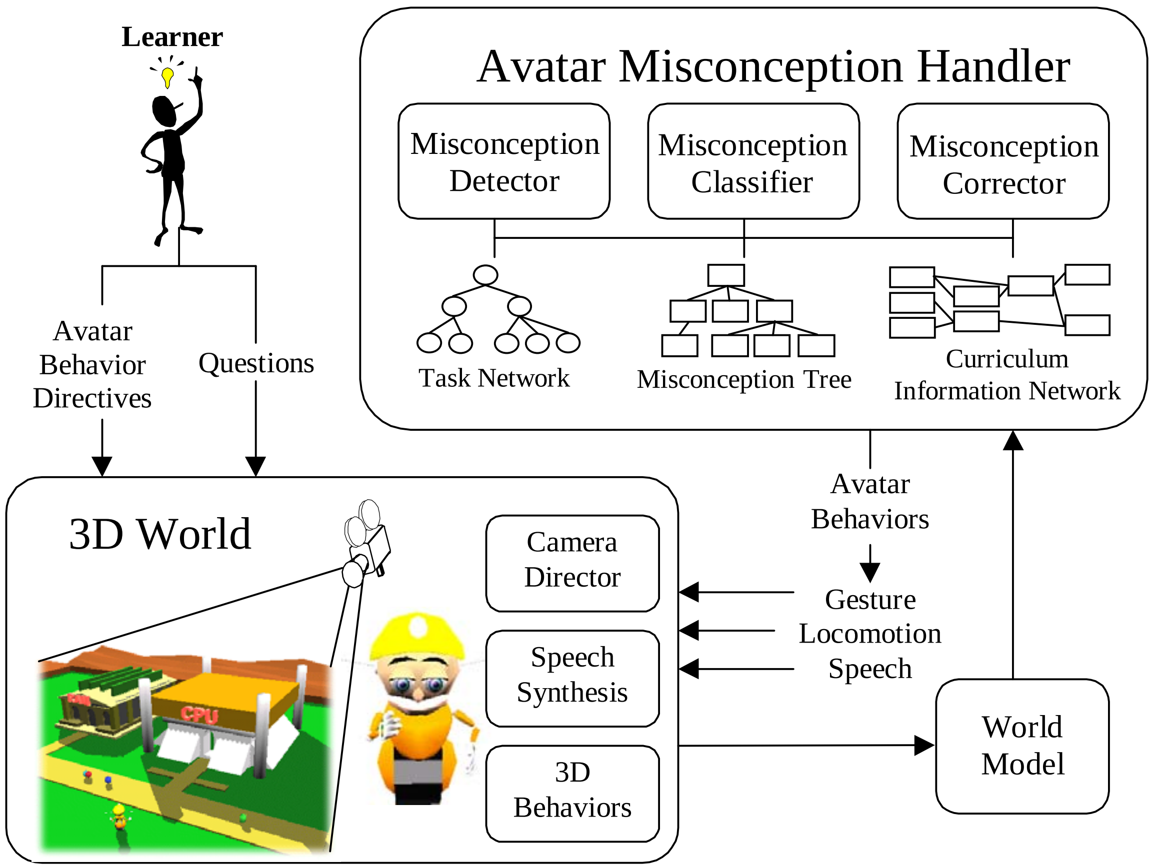


Figure 2. The avatar-based misconception detection and correction architecture

manipulate the avatar through the world to transport data packets and manipulate hardware components such as the ALU, decoder, and registers. After the avatar describes the problem, the misconception detector employs a goal-decomposition planner to create a hierarchical *task network* representing the potentially correct problem-solving actions to be taken by the learner. Task networks are graph structures whose nodes represent actions at varying levels of detail, organized in a hierarchy of time-ordered sequences. At the bottom of the hierarchy are primitive tasks whose actions require no further task decomposition. Each action specification encodes information about the type of action, the actors involved, the objects affected, and potential subsequent actions.

As the student performs actions through the avatar in the environment, the misconception detector tracks her behaviors by traversing the leaves of the task network it generated. By inspecting the action types of active action specification nodes against student-driven actions performed by the avatar in the world, the misconception detector classifies each action she performs as either critical or non-critical. Actions are considered *critical* if they enact significant changes to the problem state. For example, in CPU CITY, critical actions include attempts to pick up and deposit objects, e.g., data packets, to pass through a portal from one architectural component to another, and to manipulate a device, e.g., pulling a lever of the ALU. Periods of latency that exceed a task-specific time limit are also considered critical actions. These are used to detect misconceptions in which the student does not know what course of action to follow next and "freezes." Example *non-critical actions* in CPU CITY include turning the avatar slightly to the left or walking forward (but not through a portal).

Efficiently managing classification of actions is essential to the intention monitoring enterprise. First, to combat the problem of enumerating the full set of non-critical actions, the misconception detector polls the world model for critical actions and considers all other actions to be non-critical. Second, in learning environments that provide instruction about tasks of any complexity, multiple solution paths are possible. The misconception detector must therefore exploit a representation that is sufficiently flexible to accommodate more than one solution. Hence, the misconception detector generates task networks that are lattices of branching, partially ordered task specifications. When the student embarks on a solution that consists of a particular order of atomic actions, the misconception detector tracks her activities by traversing the task network via the particular solution path she adopts. It produces one of three outcomes: (1) if the current action is deemed non-critical, intention monitoring continues; (2) if the current action is deemed critical and correct, the misconception detector advances the action task node specification to the appropriate successors; (3) if the current action is critical but incorrect, the misconception detector invokes the misconception classifier to classify a potential misconception.

3.2 Classifying Misconceptions

When the student performs a critical action that is sub-optimal, the misconception classifier determines the type of misconception the student may have about the subject matter. Rather than invasively probing the student with questions, the misconception classifier exploits the knowledge about the student's problem-solving activities to make its diagnosis. To do so, it first inspects (1) the active task specification nodes[1] in the task network and (2) the specific action performed by the student Next, it searches through a misconception tree to determine the most specific misconception that is relevant to the student's sub-optimal actions. The *misconception tree* represents the most common misconceptions about procedural knowledge and the most common misconceptions about conceptual knowledge that may induce procedural missteps. It encodes a decision tree, where the children of each node represent specialized categories of misconceptions. At each level, the most salient problem-solving actions performed by the student and key environmental features of the 3D world are used to distinguish different categories of misconceptions. Beginning at the root, the classifier traverses the tree as deep as it can to determine the most specific misconception class applicable to the current situation.

To illustrate, suppose a student interacting with the avatar of CPU CITY has been given "programming exercise" in which she must retrieve the next microcode instruction. To do so, she must pull the Load lever in the RAM. She directs the avatar to pull the lever but has neglected to put the address in the RAM input register. Because pulling a lever is a critical action but does not successfully advance the current goal to a legitimate subsequent goal in the task network, the misconception detector signals a misconception and invokes the misconception classifier. The classifier searches through a misconception tree (Figure 3). The first decision in the tree is to determine whether the physical location of the avatar is correct with respect to the current goal. Since she is currently in the RAM, a misconception about location (*incorrect environment*) is ruled out and the classifier turns to potential inappropriate actions taken in the correct location (*rationale absent*). Because her actions have been tracked by the misconception detector, it is known that the offending behavior involved manipulating devices in the RAM, in this case, the Load lever (*incorrect lever*). Finally, by inspecting the task tree, the classifier determines that a lever pull action is currently inappropriate but will be appropriate soon; the student has skipped over an intermediate prerequisite step, a common problem in learning procedural tasks (*step skip*). Because a leaf in the tree has now been reached, the most specific category has been identified.

3.3 Correcting Misconceptions

After the student's most likely misconception has been identified, the misconception classifier invokes the misconception corrector. Given the specific category of misconception and its contextually instantiated arguments, the corrector indexes into a *curriculum information network* (CIN) [24] that encodes misconception correction topics and the prerequisite relations that hold between them. For example, given the *step skip* category identified above and the specific arguments (*RAM-load-attempt*), the corrector for the CPU CITY avatar indexes into the CIN and directs the avatar to provide verbal advice on a particular topic. A template associated with the selected topic is instantiated with lexicalizations of arguments from the current situation and the avatar is directed to provide verbal advice. In this case, the avatar informs the student, "You're skipping a step. You

[1] Multiple task specification nodes may be active because of multiple (alternate) paths through the task network to achieve a particular goal.

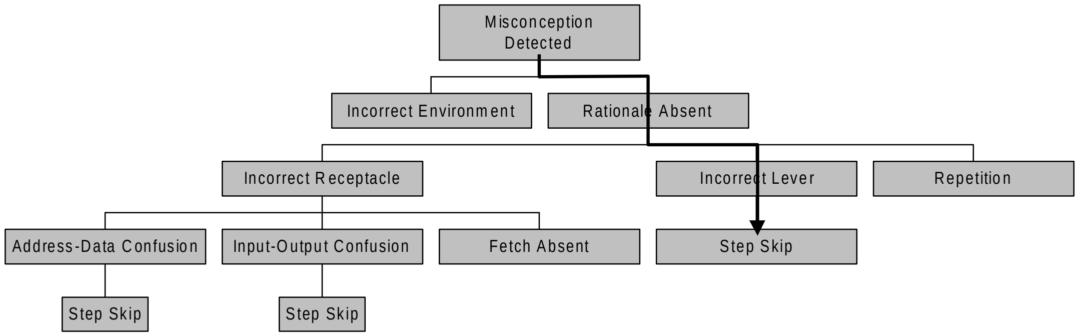


Figure 3. Example traversal of the misconception tree for CPU CITY

forgot to *X*," where *X* is instantiated here as, "put the address in the RAM input register." The strings are annotated with prosodic markups, passed to the speech synthesizer, and then spoken by the avatar.

Misconceptions are further corrected with two student-initiated question-asking techniques. If the student asks for further assistance by pressing a "help" button (Figure 1), the corrector executes the following three-step algorithm. (1) The corrector examines the students recent actions, the active task action specifications nodes in the task network, and the world model to index into the CIN. (2) It inspects an *overlay student model* [7] associated with the CIN to assess the student's prior exposure to the concept(s) discussed in the selected CIN node. (3) If the prior exposure is limited, it directs the avatar to provide a general (abstract) explanation of the relevant concepts; if there has been some degree of prior exposure, the avatar will be instructed to provide more specific assistance; if the student has been exposed to the current material multiple times and is still experiencing difficulty, the avatar will offer to perform the task for the student and explain it using pedagogical agent demonstration techniques [1,15,21]. Students may also request additional assistance by asking specific questions via a pop-up question-asking interface. If they request information about a particular topic, the corrector performs a topological sort of overlay CIN nodes to determine prerequisites of the selected concept. It then directs the avatar to provide the necessary background information and addresses the question.

4. A Lifelike Avatar for the CPU CITY Learning Environment

The misconception framework has been implemented in WHIZLOW, a lifelike avatar who inhabits the CPU CITY 3D learning environment testbed for the domain of computer architecture for novices.[2] CPU CITY's 3D world represents a motherboard housing nearly 100 3D models that represent the principal architectural components including the RAM, CPU, and hard drive. Students are given "programming" tasks in which they direct through the virtual computer. The avatar's operators used to generate task networks to track students' problem-solving activities range from operators for picking up and depositing data, instruction, and address packets to operators for interacting with devices that cause arithmetic and comparison operations. Its misconception classifier handles a broad range of misconceptions including incorrect locations for operation attempts, procedural sub-task repetitions and step skips, inappropriate device manipulations where pre-conditions have not been satisfied, and confusions between addresses and data and between input and output. The avatar's misconception corrector addresses these misconceptions by employing a CIN with more than 60 nodes. Altogether, the misconception handler, avatar behavior generator, and the CPU CITY learning environment consists of approximately 60,000 lines of C^{++} and employs the OpenGL graphics library for real-time 3D rendering.

The avatar has been subjected to a number of formative studies with more than 40 subjects interacting with WHIZLOW in CPU CITY. Most recently, the misconception framework has been investigated with a focus group study with 7 students, each of whom interacted with WHIZLOW for an

[2] The current implementation runs on Pentium II 300 MHz machines, with 64 MB of memory and an 8MB SGRAM Permedia2 OpenGL accelerator at frame rates between approximately 10-15 FPS. WHIZLOW's speech is synthesized with the Microsoft Speech SDK 3.0. Generating speech for a typical sentence requires approximately $^1/_8$ second, which includes the time to process prosodic directives.

hour to work on 5-7 programming exercises. To help the students feel comfortable, the experimenter encouraged them to pose questions to the avatar frequently. Subjects' experience interacting with WHIZLOW suggests that the task network representation is sufficiently expressive to enable the avatar to comment effectively on their activities in the world. In general, subjects were very pleased with his responses to their questions. Even though the avatar exhibited awkward movements at times, they found him extremely friendly and likeable. Because task network nodes encode preconditions on actions and their relationships with devices, they enabled the pedagogical planner to note students' problem-solving difficulties. The granularity of task networks appeared to be at approximately the appropriate level. If it were any lower, the agent would have made comments about low-level details such as micro-manipulation by subjects of the joystick, an activity with which they experienced no problems. In contrast, if the granularity were any higher the misconception detector would be unable to know where the student should be directing her avatar, what devices she should be manipulating, or how she should be performing the task.

Perhaps most critically, students interacting with avatars driven by the misconception framework learn *why* actions should be performed. First, they learn how their actions relate to constraints on the operation of the devices they operate. For example, subjects interacting with CPU CITY learned that RAM must be accessed with a specific address by being required to obtain a value from memory, but being unable to do so without specifying a particular address for it. Second, students learn the consequences of their actions through explanations provided by the agent. For example, when one subject was attempting to obtain a data packet from the RAM and a page fault occurred, WHIZLOW explained to him that he needed to go instead to the hard drive because the data sought after by the student had in fact been stored on there rather than in the RAM.

5. Conclusions and Future Work

The avatar-based misconception framework offers much promise for coupling 3D learning environments with procedurally-oriented tasks. By addressing misconceptions in the context of problem-solving, corrections offered by the avatar at these junctures—particularly by an engaging lifelike avatar—can be readily assimilated. By detecting students' misconceptions via tracking task networks, classifying misconceptions via traversing misconception trees based on features of the task and physical characteristics of the 3D learning environment, and correcting misconceptions via directing the avatar to deliver topical advice based on prerequisites in a CIN, avatars can serve as effective tools for addressing misconceptions. Three key directions for future work are particularly intriguing. First, the misconception framework currently makes a single-fault assumption. If in fact a student has multiple misconceptions about the domain—this is frequently the case, particularly for novices—the framework now can only detect and correct one misconception at a time. Addressing multiple misconceptions is critical for deploying these technologies. Second, the knowledge engineering involved in designing task tree operators, misconception trees, and CINs is substantial. Accelerating the creation of these knowledge structures and ensuring they remain mutually consistent presents non-trivial challenges. Third, conducting large-scale empirical studies in even more complex 3D worlds will shed considerable light on the most effective misconception detection and correction techniques. We will be exploring these issues in our future research.

Acknowledgements

The authors gratefully acknowledge William Bares for his work on the cinematography planner, Dennis Rodriguez for his work on the CPU CITY interface, and the members of the 3D modeling and animation team (Tim Buie, Mike Cuales, and Rob Gray), which was led by Patrick FitzGerald. Thanks also to Francis Fong for his work on the animation tools and to Charles Callaway, Brent Daniel, and Gary Stelling for comments on an earlier draft of this paper. Support for this project is provided by the National Science Foundation under grants CDA-9720395 (Learning and Intelligent Systems Initiative) and IRI-9701503 (CAREER Award Program), the North Carolina State University IntelliMedia Initiative, and Novell, Inc.

References

[1] André, E., Rist, T., & Müller, J. (1998). Integrating reactive and scripted behaviors in a life-like presentation agent. In *Proceedings of the Second International Conference on Autonomous Agents* (pp. 261-268). Minneapolis: ACM Press.

[2] Bares, W., Zettlemoyer, L., Rodriguez , D., & Lester, J. (1998). Task-sensitive cinematography interfaces for interactive 3D learning environments. In *Proceedings of the Fourth International Conference on Intelligent User Interfaces* (pp. 81-88). San Francisco: IUI '98.

[3] Bares, W., Zettlemoyer, L., & Lester, J. (1998). Habitable 3D learning environments for situated learning. In *Proceedings of the Fourth Int'l Conference on Intelligent Tutoring Systems* (pp. 76-85). Berlin: Springer-Verlag.

[4] Bates, J. (1994). The role of emotion in believable agents. *Communications of the ACM*, 37, 122-125.

[5] Blumberg, B., & Galyean, T. Multi-level direction of autonomous creatures for real-time virtual environments. In *Proceedings of SIGGRAH '95* (pp. 47-54). New York: ACM Press.

[6] Brown, J., & Burton, R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, 2, 155-191.

[7] Carr, B., & Goldstein, I. (1977). Overlays: A theory of modeling for computer-aided instruction. Massachusetts Institute of Technology, Artificial Intelligence Laboratory, AI Memo 406.

[8] Cassell, J. (in press). Embodied conversation: Integrating face and gesture into automatic spoken dialogue systems. In S. Luperfoy (Ed.), *Automatic Spoken Dialogue Systems*. Cambridge, MA: MIT Press.

[9] Chu-Carroll, J., & Carberry, S. (1994). A plan-based model for response generation in collaborative task-oriented dialogues. In *Proceedings of the Twelfth National Conference on Artificial Intelligence* (pp. 799-805). Seattle, Washington: American Association for Artificial Intelligence.

[10] Dillenbourg, P., Jermann, P., Schneider, D., Traum, D., & Buiu, C. (1997). The design of MOO agents: Implications from an empirical CSCW study. In *Proceedings of the Eighth World Conference on Artificial Intelligence in Education* (pp. 15-22). Kobe, Japan: IOS Press.

[11] Eliot, C., & Woolf, B. (1996). A simulation-based tutor that reasons about multiple agents. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence* (pp. 409-415). Portland, Oregon: American Association for Artificial Intelligence.

[12] Hietala, P., & Niemirepo, T. (1998). The competence of learning companion agents. *International Journal of Artificial Intelligence in Education*. 9(3-4), 178-192.

[13] Hill, R., & Johnson, W. L. (1995). Situated plan attribution. *Journal of AI in Education*, 6, 35-66.

[14] Lester, J.C., Converse, S.A., Kahler, S. E, Barlow, T., Stone, B.A., & Bhogal, R. (1997). The persona effect: Affective impact of animated pedagogical agents. In *Proceedings of CHI '97 (Human Factors in Computing Systems)* (pp. 359-366). Atlanta: ACM Press.

[15] Lester, J., Zettlemoyer, L., Grégoire, J., & Bares, W. (in press). Explanatory lifelike avatars: Performing user-designed tasks in 3D learning environments. To appear in *Proceedings of the Third International Conference on Autonomous Agents*, Seattle, Washington.

[16] Marsella, S., & Johnson, W.L. (1998). An instructor's assistant for team-training in dynamic multi-agent virtual worlds. In *Proceedings of the Fourth Int'l Conference on Intelligent Tutoring Systems* (pp. 464-473). Berlin: Springer-Verlag.

[17] McCoy, K. (1989-90). Generating context-sensitive responses to object-related misconceptions. *Artificial Intelligence*, 41, 157-195.

[18] Mengelle, T., De Léan, C., & Frasson, C. (1998). Teaching and learning with intelligent agents: Actors. In *Proceedings of the Fourth Int'l Conference on Intelligent Tutoring Systems* (pp. 284-293). Berlin: Springer-Verlag.

[19] Paiva, A., & Machado, I. (1998). Vincent, an autonomous pedagogical agent for on-the-job training. In *Proceedings of the Fourth Int'l Conference on Intelligent Tutoring Systems* (pp. 584-593). Berlin: Springer-Verlag.

[20] Piaget, J. (1954). *The construction of reality in the child*. New York: Basic Books.

[21] Rickel, J., & Johnson, W. L. (1997). Intelligent tutoring in virtual reality: A preliminary report. In *Proceedings of the Eighth World Conference on AI in Education* (pp. 294-301). Kobe, Japan: IOS Press.

[22] Ritter, S. (1997). Communication, cooperation, and competition among multiple tutor agents. In *Proceedings of the Eighth World Conference on AI in Education* (pp. 31-38). Kobe, Japan: IOS Press.

[23] Wang, W., & Chan, T. (1997). Experience of designing an agent-oriented programming language for developing social learning systems. *Proceedings of the Eighth World Conference on AI in Education* (pp. 7-14). Kobe, Japan: IOS Press.

[24] Wescourt, K., Beard, M., & Barr, A. (1981). Curriculum information networks for CAI: Research on testing and evaluation by simulation. In P. Suppes (Ed.), *University-level Computer-Assisted Instruction at Stanford: 1968-1980* (pp. 817-839). Stanford, CA: Stanford University Press.