Explanatory Lifelike Avatars: Performing User-Centered Tasks in 3D Learning Environments

James C. Lester, Luke S. Zettlemoyer, Joël P. Grégoire, and William H. Bares

Multimedia Laboratory
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206

{lester,lszettle,jgregoi,whbares}@eos.ncsu.edu

ABSTRACT

Because of their multimodal communicative abilities and strong visual presence, animated pedagogical agents offer significant promise for 3D learning environments. We describe a new class of animated pedagogical agents, *explanatory lifelike avatars*, which can perform user-designed tasks in rich 3D worlds. By generating task networks to perform student-designed tasks, an avatar task planner constructs and interprets action specifications that it then interprets within the geometries of the 3D environment to generate navigational, manipulative, and verbal behaviors. Filmed by a narrative camera planner in the 3D world, the avatars perform students' tasks and accompanies them with running verbal explanations in realtime. The explanatory lifelike avatar framework has been implemented in a full-scale avatar for the CPU CITY learning environment, a 3D learning environment for the domain of computer architecture and systems for novices. To investigate the effectiveness of this approach, a novel four-way comparative usability study was conducted with an "agentless" world, a disembodied narrator, a mute lifelike avatar, and a full-scale explanatory avatar. Results of the study suggest that explanatory lifelike avatars hold much promise for learning environments.

Keywords

Lifelike agents, pedagogical agents, synthetic agents, knowledge-based learning environments.

1. INTRODUCTION

Recent years have witnessed rapid progress in synthetic animated agents that exhibit increasingly sophisticated lifelike qualities [5,6,8,9,13,20]. While much of this work has addressed core issues in agents for communication and entertainment, we have also begun to see significant results in animated pedagogical agents for learning environments [1,12,16,19]. By combining lifelike behaviors with pedagogical strategies, animated pedagogical agents explain complex concepts and provide advice to students throughout problem-solving episodes. Because of their compelling visual presence, these agents offer much promise for learning effectiveness and motivation. One of the most intriguing possibilities offered by these developments is the potential of creating rich, learner-centered 3D environments populated by lifelike animated agents. If students could design tasks to be performed in a 3D world and then observe the tasks playing out as the agents explained the tasks, students would be afforded significant opportunities for understanding the complexities of the simulated system. Microworlds have become a staple of the knowledge-based learning environments community, both in research laboratories and in commercially available systems. However, these kinds of systems typically have very limited explanatory capabilities: while students can affect microworld simulations, microworlds typically lack the ability to clearly explain events in a compelling manner.

To address these issues, we have developed a framework for creating *explanatory lifelike avatars* in 3D learning environments. In this framework, students investigate complex simulated systems by interactively instructing their avatars to undertake tasks in the 3D world. In response, the avatars' behavior planners construct and execute explanatory demonstrations in which they carry out the tasks and explain their actions. By exploiting a task constructor that builds a task network that is interpreted in the context of the 3D world, the avatars perform student-specified tasks and coordinate physical behaviors with verbal explanations in realtime.

The explanatory lifelike avatar framework has been implemented in an animated pedagogical agent, WHIZLOW, who inhabits the CPU CITY 3D learning environment (Figure 1). The CPU CITY environment provides technical novices (i.e., non-technical students) with a virtual computer cityscape housing a CPU, RAM, hard disk, and the buses connecting them to teach them about the fundamentals of computer architecture. Using a high-level programming language to issue their task specifications, students direct WHIZLOW to perform tasks within the virtual computer. To investigate the effectiveness of this approach, an informal 4-way comparative usability study was conducted with (1) an "agentless" simulation in the 3D world, (2) a disembodied narrator, (3) a mute lifelike avatar who carried out the students' tasks in the 3D world, and (4) an explanatory lifelike avatar with a full complement of demonstration and explanation capabilities. Results of the study are encouraging and suggest that explanatory lifelike avatars hold much promise for learning environments.

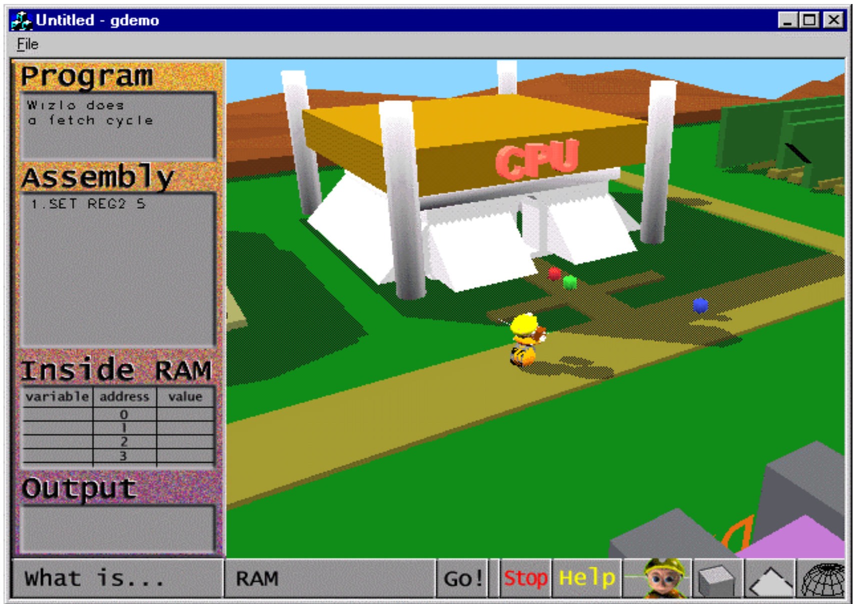


Figure 1: The WhizLow Lifelike Avatar in the CPU City 3D Learning Environment

2. Lifelike Pedagogical Agents and Explanatory Avatars

Because of lifelike pedagogical agents' abilities to combine sophisticated communicative functionalities with engaging personae [1,12,16,19], they can take advantage of humans' inherent propensities to anthropomorphize software [15] and play a central role in students' problem-solving activities. We have also begun to see the results of rigorous experiments indicating they may contribute substantially to learning effectiveness [11] and students' positive perception of learning experiences [1,10].

While progress on lifelike pedagogical agents has been significant, its focus to date has been on creating animated agents that observe students' problem-solving activities in a kind of "over-the-shoulder" mode and then provide explanations and advice to support learning. An alternate and complementary approach to introducing lifelike pedagogical agents into learning environments is as lifelike avatars that can embody tasks to be performed in simulated worlds. Students could explore these environments by constructing high-level task specifications and then observing their avatar interacting with the world to perform their tasks. While we have experimented extensively with "direct control" lifelike avatars that are driven directly by students with a joystick [4], it appears that students often become mired in the details of navigation and manipulation. In contrast, by enabling students to construct tasks and request lifelike avatars to carry them out, they can attend to the critical concepts in the domain. Ideally, the avatar would not only carry out the task but verbally explain his behaviors as he did so.

If successfully designed, this class of animated pedagogical agents could create rich learning interactions for a broad range of domains. For example, in human anatomy and physiology, students could direct lifelike agents representing corpuscles to navigate through the circulatory system and pick up and deposit oxygen to keep the body vital. In chemistry, students could direct lifelike avatars representing molecules to form and break bonds to synthesize a variety of compounds. In physics, they could direct lifelike avatars representing electrons to travel through circuits and electric fields *en route* to magnets and batteries, attracting and repelling fellow electrons and inducing forces along the way.

Given their dual function of pedagogy (for learning effectiveness) and exhibiting a strong visual presence (for motivation), lifelike explanatory avatars should satisfy the following design criteria:

- *Student-Constructed Task Design:* Students should be able to explore the complexities of the simulated world by constructing task specifications. While other animated pedagogical agents projects provide rich explanation capabilities such as those exhibited by the Steve (Soar Training Expert for Virtual Environments) agent [16], the PPP persona [1], Herman the Bug [19], and Cosmo [12], their explanations are in the form of problem-solving advice. For example, Steve has the most sophisticated demonstration facility ever developed in lifelike agents, but its focus is on demonstrations of system-determined tasks rather than tasks created by students. Nevertheless, because demonstrating student-designed and agent-designed tasks both involve the same set of communicative issues, we can bring to bear many of the lessons learned in the Steve project, e.g., employing a plan-based representation of task knowledge and enabling agents to successfully inhabit a 3D world, in designing the explanatory avatar lifelike framework.
- *Embodied Lifelike Explanation:* The traditional approach to introducing explanation facilities into learning environments is via text-based dialogues [14]. However, given the motivational benefits of lifelike agents [1,10], "embodying" explanations in onscreen personae entails creating avatars that are (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 carry out students' tasks, and (3) adept at coordinating their physical behaviors with a running verbal explanation that is tightly coupled to the task being performed.

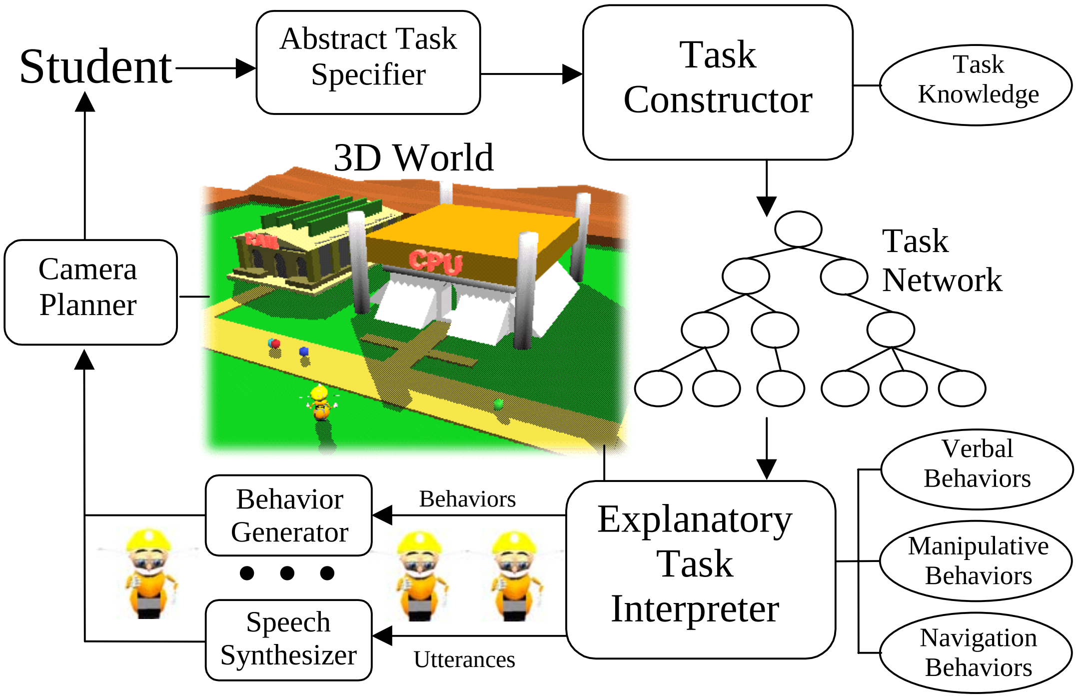


Figure 2: The Explanatory Lifelike Avatar Architecture

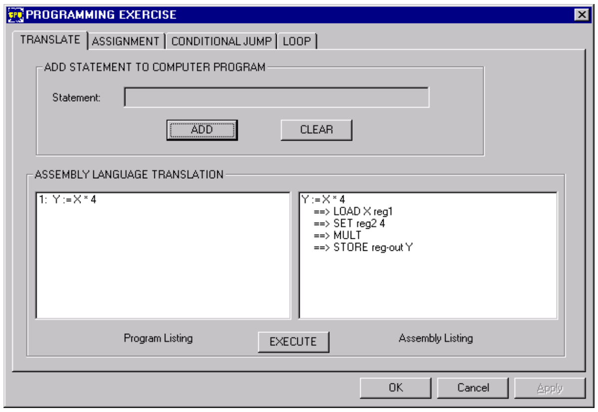
- *Abstract Task Specifications:* Explanatory lifelike avatars can be an effective vehicle for promoting learning if they can enable students to focus their attention on the task at hand. This has two important implications: (1) Rather than requiring students to create low-level specifications in a difficult-to-master control language, the specifications should be at a relatively abstract level and be expressed in constructs that are tightly coupled to the domain. (2) The avatar behavior sequencing mechanisms should themselves attend to the low-level details and free students to focus on more central considerations. In the same manner that the BodyChat avatar [20] generates animated communicative behaviors for turn-taking, facial expressions, and back-channel feedback in conversations, lifelike avatars for learning environments should attend to all of the low-level physical details of route planning and device manipulation.

3. Explanatory Lifelike Avatars for 3D Environments

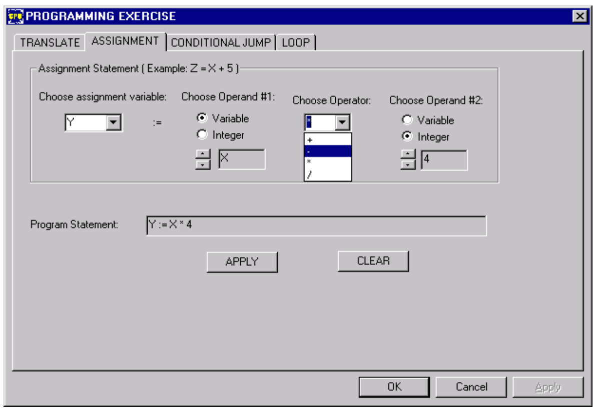
In the explanatory lifelike avatars framework (Figure 2), students interact with a 3D learning environment representing a complex device or physical system. Each artifact in the world represents a component of the virtual device or system, and actions in the world represent activities and functionalities, which are driven by a 3D simulation. For example, in the CPU City learning environment (Figure 1), buildings represent the components such as the CPU and devices within these buildings represent sub-components such as the registers and the ALU. Computations in CPU City are visually represented by data and address packets traveling between components, and a virtual machine performs all arithmetic, logical, and memory access operations. To interact with their explanatory lifelike avatars, students design tasks for their avatars to perform by constructing high-level task specifications with an *abstract task specifier*. Once task specification begins, the avatar's behaviors are generated by the following three phase process:

1. **Task Specification and Construction:** Tasks are described by students in terms of abstract operations they would like to see play out in the simulated world. After the task has been completely specified, the specifications are sent to a *task constructor*. By exploiting a rich representation of task knowledge, the task constructor's goal-decomposition planner creates a task tree whose leaves are action specifications for the avatar to perform. To maintain a separation of abstract task knowledge and the details of executing low-level behaviors with context-sensitive geometric considerations, the leaves of the task tree (the action specifications) are recommendations for avatar actions which will then be interpreted within the physical realities of the 3D world.
2. **Explanatory Task Interpretation:** The *explanatory task interpreter* traverses the action specifications produced in Step (1), each of which symbolically (though not geometrically) represents the locations in the 3D world and the artifacts to be manipulated. The interpreter uses this knowledge to determine the *navigation*, *manipulation*, and *verbal* behaviors. Navigation behaviors attend to the intricacies of route planning, manipulation behaviors are interpreted in the context of the particular devices that are being manipulated, and verbal behaviors are sequenced to create the accompanying narration.

The physical behaviors (locomotion and manipulatives) generated in Step (2) are passed to a *3D behavior generator*, and the narrative behaviors are passed to a speech synthesizer. As the avatar performs the physical actions in the world, a *narrative cinematography planner* [3] determines the shots, pans, and zooms for the virtual camera that will "film" the avatar's behaviors. The net effect of these activities is a rich, immersive



(a) Main screen: inserting an assignment statement.



(b) Assignment screen: creating assignment statements.

Figure 3: The Task Specification Interface of the CPU City Learning Environment

experience—in our case, these are conducted in onscreen 3D rather than in headmounted VR—in which the student constructs a variety of tasks and observes an engaging lifelike character performing her tasks directly in the world.

3.1 Task Specification and Construction

Simultaneously addressing the pedagogical needs of students and the behavior specification needs of avatars requires a dual representation of task knowledge. To enable students to construct high-level specifications, task knowledge must be represented at a high level of abstraction and be expressed in domain-specific constructs that are germane to the student's learning experience. In contrast, to enable the avatar planner to generate physical and narrative behaviors, the representation of task knowledge must be expressed in a formalism that permits the task interpreter to devise appropriate navigational, manipulative, and verbal behaviors. Satisfying these two requirements is accomplished by providing a dual representation consisting of (1) a *student-centered* representation that is tightly coupled to the domain, and (2) an *avatar-centered* representation that can be interpreted in the geometries of the 3D world to create avatars' demonstrative and explanatory behaviors.

This first phase of generating the avatar's behaviors consists of task specification, translation, and construction. During *task specification*, users design their creating expressions in the student-centered representation. Next, during *task translation*, the student-centered representation is translated to high-level goals in the avatar-centered representation. Finally, during *task construction*, the task planner generates action specifications for the avatar by employing a goal-decomposition planner [17].

Task Specification. To ensure that students can easily construct specifications for potentially complex tasks, it is critical that the task specification mechanism have three properties. First, its language must directly employ domain concepts and not require the student become acquainted with some complex formalism. For example, the domain of the CPU City learning environment is introductory computer architecture and systems, so its task specification language should employ central concepts in system fundamentals. It therefore provides students with a very high-level procedural programming language with constructs drawn from languages introduced in computer literacy courses. Second, the student-centered language should be generative, i.e., it should allow students to create expressions, impose particular orders on them, and have them generate a broad range of behaviors. For example, even though CPU City's task specification language is simple, it enables students to design a variety of computations on the virtual machine. Finally, the task specification interface should be designed so that students build up statements (rather than writing them from scratch) to avoid syntax errors. CPU City's task specifier interface, for example, provides a comprehensive set of menus for building statements in the high-level language (Figure 3).

Task Translation. As noted above, task construction consists of (1) translating the task specification expressed in the student-centered language to an equivalent specification of high-level goals in the avatar-centered language, and (2) planning the avatar's action specifications from the high-level goals. The translation process is accomplished with a straightforward mapping that produces expressions that can be used by the underlying simulation of the learning environment. For example, in CPU City, the high-level statements created by the student in the programming language are translated to assembly language statements that run on CPU City's virtual machine.

Task Planning. After translation is complete, the task constructor employs a goal-decomposition planner to build the avatar's action specifications. All task knowledge at this level is encoded with a procedural network representation [17] whose nodes represent goals at varying levels of detail. At the leaves of the hierarchy are avatar action specifications that will be interpreted in the succeeding phase. Temporal dependencies between actions are included in the representation to guide the task interpreter. Planning consists of instantiating nodes and recursively sub-goaling until action specifications are encountered. For example, a student-specified task in CPU City for converting temperature from Celsius to Fahrenheit is first translated to the virtual assembly language, and each statement in the assembly language becomes a high-level goal for the avatar (Figure 4). The task constructor then creates a complete plan with which the avatar will achieve each goal by navigating through the 3D world of the virtual computer, appropriately picking up and depositing data and address packets and interacting with devices in the CPU, memory, and the hard drive along the way.

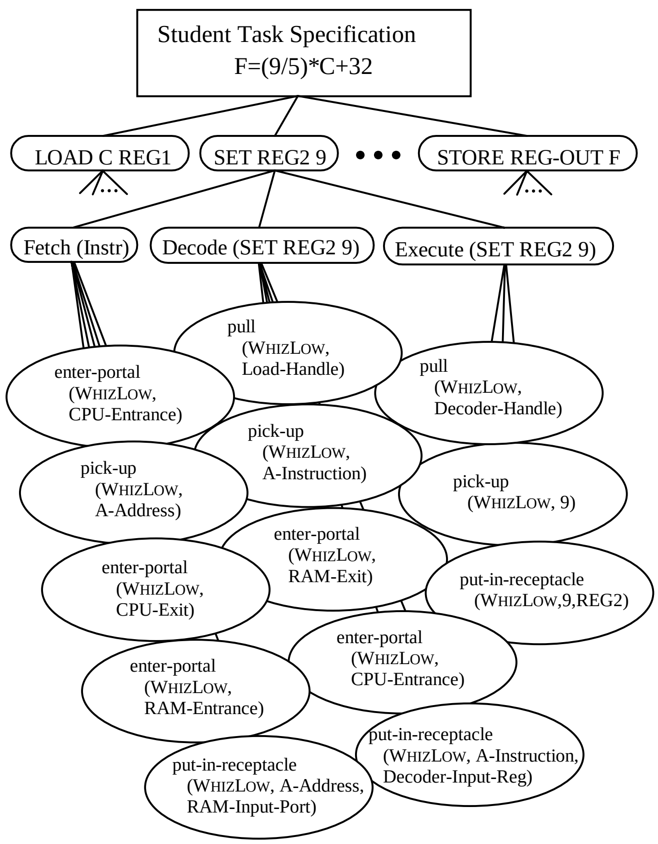


Figure 4: Fragment of a Task Network for the CPU City Avatar

3.2 Explanatory Task Interpretation

After task planning is complete, the action specifications at the leaves of the task network must be interpreted in the 3D world. It is the interpreter's job to inspect each action specification created above to determine the navigation, manipulation, and narrative behaviors that the avatar will exhibit in the world to carry out and explain the student's task. Action specifications include the knowledge typically found in plan operators such as the type of action to be performed, the structures and locations in which they should be performed, the artifacts and devices relevant to the action, and the effects they have on the world. For example, one task network generated for the CPU City avatar to enable it to perform the Celsius-Fahrenheit conversion mentioned above caused the planner to generate 67 nodes, including the action specifications shown in Figure 5.

Interpreting Task Specifications. To interpret action specifications, the interpreter must first create navigation behaviors that are compatible with the geometries of the structures and artifacts in the 3D world. Computing navigation entails determining where the avatar is located prior to the current action specification's interpretation, where it needs to be to accomplish the current specification, and the path it should travel if navigation is called for. Second, it must create manipulative behaviors in which the avatar interacts with the objects and devices to accomplish particular steps of the task. For example, in the CPU City learning environment, the avatar is called upon to pick up data and address packets, drop them off at particular locations, and interact with specific computational devices. Third, it must create narrative behaviors which the avatar will use to explain its navigation and manipulatives. Narrative behaviors are represented as explanatory sentential templates. Each narrative behavior is indexed by action-specification types and annotated with prosodic markings for the speech synthesizer. At runtime, the interpreter instantiates them with lexical items associated with the values in the action specifications. For each action specification *S*, the interpreter builds explanatory demonstration sequences as follows:

1. **Determine focus objects *F* of *S*.** Inspect *S* to identify locations and artifacts in the world relevant to *S*, such as the *portal* in `Action-Specification-869` (Figure 5).
2. **Determine navigation behaviors *N*.**
   (a) Determine locations L_f for all focus objects *F*. By examining the geometries of the 3D world model, find all destination locations.
   (b) Determine avatar's current location L_a.
   (c) If $L_a \neq L_f$ then compute *N* with the Bézier navigation planner described below.
3. **Determine manipulative behaviors *M*.**
   (a) Determine objects *O* to be manipulated in *S*, such as the *receptacle* (the Decoder-Input-Register) in `Action-Specification-870` (Figure 5).
   (b) Inspect the *action-type* of *S* to identify the manipulative behaviors *M* to apply to *O*.
4. **Determine verbal behaviors *V*.**
   (a) Determine *navigative* narratives V_n that explain the navigation behaviors that the avatar will perform by inspecting *N*.
   (b) Determine the *manipulative* narratives V_m that explain the manipulation behaviors that the avatar will perform by inspecting *M*.
5. **Construct the explanatory demonstration sequence (V_n, *N*, V_m, *M*).**

Bézier Navigation Planning. While generating simple manipulative behaviors is relatively straightforward, creating navigation behaviors that are *believable* is non-trivial because navigation trajectories must appear natural. To do so, the navigation behavior generator plans routes as follows: Given the avatar's current location and its target destination as determined above, the navigation planner first invokes A* [7] 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 Bézier spline that interpolates the discretized path from the avatar's current location, through each successive control point, to the target destination.[1] For example,

[1] A Bézier spline is a sequence of polynomial curves of degree one less than the number of control points used, e.g., three points generate a parabola. Bézier curves have three useful properties: (1) they always pass through the first and last control points, and (2) the tangent to the curve at an endpoint is along the line joining that endpoint to the adjacent control point, and (3) the curve is completely contained within the convex hull defined by its control points. To obtain continuity between multiple curve sections, we select the first two control points of the next section aligned with the last two control points of the previous curve section. To obtain a curve that passes through all of the control points, additional control

Action-Specification-869
action-type: Enter-Portal
actor: WHIZLOW
location: Computer
effect: inside(CPU)
succeeding-action: AS-870

Action-Specification-870
action-type: Receptacle-Object-Placement
actor: WHIZLOW
location: CPU
effect: holding(Decoder-Input-Register, Instruction-Packet-A)
succeeding-action: AS-871

Action-Specification-871
action-type: Lever-Pull
actor: WHIZLOW
location: CPU
effect: pulled(Decoder-Lever)
succeeding-action: AS-872

Figure 5: Sample Action Specification for the CPU CITY Avatar

when the avatar in the CPU City learning environment must navigate around a corner of a building, the Bézier planner creates a path that enables him to smoothly turn rather than taking a sharp 90° turn while still ensuring that he does not collide with the building. The Bézier approach builds on advances in the robotics community. It enables the avatar to traverse paths that appear noticeably smoother than agents employing motion path planners that use only a grid-based representation, e.g., [2].

4. An Implemented Lifelike Avatar

The explanatory lifelike avatar framework has been implemented in WHIZLOW, a lifelike avatar for the CPU CITY 3D learning environment developed in our laboratory to teach non-technical novices the fundamentals of computer architecture and systems.[2] 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. Its high-level task specification language is a procedural programming language with constructs for conditionals, assignments, and iteration. WHIZLOW can carry out student's 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, the avatar begins performing the student's task in less than a second.

To illustrate the behavior of the explanatory avatar framework, consider the following situation in a CPU CITY learning session. A student has been interacting with WHIZLOW for some time, having asked him to perform a variety of computational tasks beginning with simple assignments. She now decides she would like to see how instructions and data flow through the machine when a user executes a Celsius-Fahrenheit conversion program. To do so, she uses the task specifier to design expressions in the high-level language that instruct WHIZLOW to carry out the computations that will compute the conversion. The task constructor converts the high-level expressions to nine assembly language statements for the virtual computer, which it then translates to a series of high-level goals. Next, the task planner uses top-down goal decomposition to create a task network that guides WHIZLOW's actions. It passes each of the more than 50 action specifications to the task interpreter, which orchestrates WHIZLOW's navigation, manipulation, and verbal behaviors. These are then passed to the cinematography planner, which craft a several minute demonstration in which WHIZLOW performs the computational steps of the conversion computation.

First, WHIZLOW travels to the CPU. As the camera focuses on his face, he explains, "I decode the next instruction by pulling the decode handle" (Figure 6). The camera cuts to a wider shot and swings around to clearly depict WHIZLOW wheeling over to the control panel and pulling the lever. The load instruction packet then materializes in the CPU, and the camera view returns to WHIZLOW who notes that, "It's the *load* instruction. A load instruction allows the CPU to retrieve a value from RAM based on its address." The camera then returns to a wider shot as WHIZLOW proceeds to execute the instruction, which entails traveling to the RAM, obtaining a data packet representing the Celsius value to be converted, and returning to the CPU where he places it in a register. At each juncture, he explains his actions, frequently relating them to core concepts of computation. He then continues to perform navigation and manipulation behaviors dictated by the action specifications, all the while providing a running narrative. He finally completes the task as he explains, "The variable *F* now has the value 68 and is stored in RAM." The student then continues her exploration by devising more complex tasks such as those involving conditionals.

5. EVALUATION

To gauge the effectiveness of the explanatory lifelike avatars framework, a usability study was conducted with students interacting with WHIZLOW and the CPU CITY learning environment. The study was designed to investigate five questions: (1) Do explanatory lifelike avatars contribute to learning effectiveness? (2) Do explanatory lifelike avatars contribute to learning enjoyment? (3) What is the relative impact of their demonstrative (physical) capabilities? (4) What is the relative impact of their explanatory (verbal) capabilities? (5) Are there any differences on the learning impact of explanatory lifelike avatars that stem from students' position on the novice-expert spectrum?

5.1 Experimental Design

To answer these questions, we designed a 4-way comparative usability study in which subjects would interact with different versions of the CPU CITY 3D learning environment. In particular, we sought to isolate specific communicative modalities while keeping all other aspects of the learning experience identical. Each of the four systems shared precisely the same 3D world, the

points need to be inserted. In our 2D grid, two control points are added close to each original control point (corners) except the source and the destination. The short distance between each corner and its two control points prevent the path from colliding with obstacles. The path is finally generated by creating a series of Bézier curves between each corner.

[2]The explanatory lifelike avatar system and the CPU CITY learning environment consist of approximately 60,000 lines of C^{++}. They employ the OpenGL graphics library for 3D rendering. The system runs on Pentium II 300 MHz machines, with 64 MB of memory and 8 MB SGRAM Permedia2 OpenGL accelerators at frame rates between 10-15 fps. The avatar's speech is generated with Microsoft's Speech SDK 3.0. Generating speech for a typical sentence typically requires $^1/_8$ second, which includes time to process prosodic requests for emphasis.

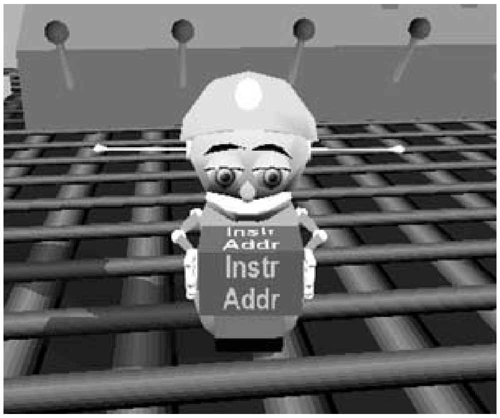


Figure 6: WHIZLOW explains decoding.

structures and objects within it, the functionality of all of the devices (e.g., the ALU, register operation, memory access), the behaviors of the data packets and address packets, the behavior of the camera "filming" the action, and the virtual machine within which all operations took place. Keeping these constant, we created four learning environments:

- **Agentless 3D World:** In the "baseline" environment, subjects specify tasks which are to be simulated in the 3D world. In these simulations, data packets and address packets traverse the same routes as if they were carried by agents, but they float by themselves to their appropriate destinations. The virtual camera follows their actions in exactly the same manner as it would in other versions of the learning environment inhabited by the avatar.
- **Disembodied Narrator: A**n off-screen speaker explains the activities of the simulations as they are specified by the students and played out in the world. The narration is generated with the verbal behavior type annotations on action specifications in task trees as described above; however, rather than being spoken by the agent, the narration is uttered by an unseen speaker using exactly the same voice.
- **Mute Lifelike Avatar:** Students specify tasks in precisely the same manner as in the other environments, and the tasks are carried out by WHIZLOW. He travels through the buildings of the 3D world, carries and deposits packets, and manipulates devices as dictated by the task tree constructed in response to the students' specifications. However, his actions are stand-alone: they are not accompanied by a running explanatory narrative.
- **Explanatory Lifelike Avatar:** Students specify tasks in the same manner as in the other environments. In response, the avatar carries out their requests in the 3D world as above. However, in addition to his physical activities, he provides a step-by-step running verbal commentary that is tightly coupled to his demonstrative actions.

To obtain a broad spectrum of learning styles and preferences, the study employed 12 subjects with varying degrees of expertise in the domain of computer architecture fundamentals. Some subjects were complete novices, others had some programming expertise, while others were experts. Subjects were equally divided between men and women. On average, each subject interacted with CPU CITY for 45 minutes. To avoid overwhelming the subjects with four versions of the learning environment (both in terms of time and conceptual overload), the study was designed so that each subject interacted with two versions of the learning environment.

Sessions proceeded as follows. First, each subject was given a brief introduction to the basics of computer architecture. This included an overview of hardware components and functionalities provided by the experimenter.[3] All subjects interacted with two versions of the learning environment: the Agentless 3D World and one other environment, either the Disembodied Narrator, the Mute Lifelike Avatar, or the Explanatory Lifelike Avatar. To control for order effects, half of the subjects interacted with the Agentless 3D World before interacting with a learning environment with more sophisticated communicative abilities, while the other half of the subjects interacted with the more advanced learning environment first. Throughout the interactions and after interacting with each environment, subjects were encouraged to comment on their learning experiences.

5.2 Results

Bearing in mind the caveat that the study was informal, the findings are encouraging:

- It seems clear that explanatory lifelike avatars can contribute significantly to learning experiences. Subjects who interacted with the Agentless 3D World and the Explanatory Lifelike Avatar environment unanimously preferred the lifelike avatar. Each commented on the clarity of communication and the extent to which the avatar helped them to grasp the complexities of the subject matter. This result seems to stem from the focus that the embodied agent visually brings to the experience, as well as the complementary information provided by his narration.
- The study suggests that lifelike avatars can contribute to both learning effectiveness and motivation. On the effectiveness side, subjects repeatedly commented about the improved clarity of their understanding by interacting with the environment with the avatar. With regard to motivation, most subjects (without prompting) commented that they very much enjoyed interacting with the avatar. They also commented it was much easier for them to maintain their attention for longer periods of time with the avatar.
- Preferences for the Explanatory Lifelike Avatar environment over the Agentless 3D World are much stronger than for either the Disembodied Narrator or the Mute Lifelike Avatar over the Agentless 3D World. While the demonstrations of the Mute Lifelike Avatar were much better received with regard to visual focus and enjoyment than the Agentless World, the verbal explanations of the Disembodied Narrator were better received with regard to improving the clarity of the activities of the simulation (as compared to the Agentless 3D World). Comments were noticeably less positive about both the Disembodied Narrator and the Mute Lifelike Avatar than about the Explanatory Lifelike Avatar environment.
- The benefits provided by the explanatory lifelike avatar depend on the expertise of the students. While students with high initial expertise commented positively about the avatar, the comments by the mid-range and novice students were even stronger. This suggests that learning environments for novices stand considerably more to gain from explanatory lifelike avatars than learning environments designed for their more advanced counterparts.

[3] Although CPU CITY includes fairly extensive explanatory capabilities which WHIZLOW provides with expressive walk-through explanations, we did not want to unknowingly bias subjects, so explanations were presented directly by the experimenters.

6. CONCLUSIONS AND FUTURE WORK

Because of their strong visual presence and clarity of communication, explanatory lifelike avatars offer significant potential for playing a central role in next-generation learning environments. We have proposed a computational model for enabling students to design tasks that are dynamically performed and explained by avatars directly in rich 3D worlds. Exploiting a task constructor that builds a task network which is interpreted in the geometries of the 3D world and "filmed" by a narrative camera planner, the avatars perform the student-specified tasks, create natural navigation behaviors, manipulate devices in the environment, and accompany their actions with verbal explanations in realtime to keep students deeply engaged in the learning environment's activities.

This work represents a promising step toward creating lifelike explanatory avatars for 3D learning environments. However, significant challenges remain, including endowing lifelike avatars with models of emotion and providing them with much more sophisticated natural language generation capabilities to increase their flexibility and clarity of expression. We will be investigating these directions in our future research.

ACKNOWLEGEMENTS

The authors gratefully acknowledge the members of the 3D modeling 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, to Dennis Rodriguez for his work on the task specifier interface, and to Stuart Towns for comments on earlier drafts 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., and Muller, J. Integrating reactive and scripted behaviors in a life-like presentation agent. In *Proceedings of the Second International Conference on Autonomous Agents* (1998), 261-268.

[2] Bandi, S., and Thalmann, D. The use of space discretization for autonomous virtual humans. In *Proceedings of the Second Int'l Conference on Autonomous Agents* (1998), 336-337.

[3] Bares, W. and Lester, J. Intelligent multi-shot visualization interfaces for dynamic 3D worlds. In *Proceedings of the 1999 International Conference on Intelligent User Interfaces.* (1999), 119-126.

[4] Bares, W., Zettlemoyer, L., and Lester, J. Habitable 3D learning environments for situated learning. In *Proceedings of the Fourth International Conference on Intelligent Tutoring Systems* (1998), 76-85.

[5] Blumberg, B. and Galyean, T. Multi-level direction of autonomous creatures for real-time virtual environments. In *Proceedings of ACM SIGGRAPH* (1995), 47-54.

[6] Cassell, J., Pelachaud, C., Badler, N., Steedman, M., Achorn, B., Becket, T., Douville, B., Prevost, S., and Stone, M. Animated conversation: Rule-based generation of facial expression, gesture and spoken intonation for multiple conversational agents. In *Proceedings of SIGGRAPH* (1994)

[7] Hart, A., Nilsson, N., and Raphael, B. A formal basis for the heuristic determination of minimal cost paths. *IEEE Trans. on Systems, Science, and Cybernetics*, 4(2) (1968), 100-107.

[8] Hayes-Roth, B. and van Gent, R. Story-making with improvisational puppets. In *Proceedings of the First Int'l Conference on Autonomous Agents* (1997), 1-7.

[9] Kurlander, D. and Ling, D. T. Planning-based control of interface animation. In *Proceedings of CHI* (1995), 472-479.

[10] Lester, J., Converse, S., Kahler, S., Barlow, S., Stone, B., and Bhogal, R. The persona effect: Affective impact of animated pedagogical agents. In *Proceedings of CHI* (1997), 359-366.

[11] Lester, J. C., Converse, S. A., Stone, B. A., Kahler, S. E., and Barlow, S. T. Animated pedagogical agents and problem-solving effectiveness: A large-scale empirical evaluation. In *Proceedings of Eighth World Conference on Artificial Intelligence in Education* (1997), 23-30.

[12] Lester, J. C., Voerman, J., Towns, S., and Callaway, C. Deictic Believability: Coordinating Gesture, Locomotion, and Speech in Lifelike Pedagogical Agents. *Applied Artificial Intelligence*. (1999) In press.

[13] Loyall, B., and Bates, J. Personality-rich believable agents that use language. In *Proceedings of the First International Conference on Autonomous Agents* (1997), 106-113.

[14] Moore, J. *Participating in Explanatory Dialogues.* (1995) MIT Press.

[15] Reeves, B. and Nass, C. The Media Equation: How People Treat Computers, Television and New Media Like Real People and Places. (1998) CSLI, New York.

[16] Rickel, J. and Johnson, L. Animated agents for procedural training in virtual reality: Perception, cognition, and motor control. *Applied Artificial Intelligence*. (1999) In press.

[17] Sacerdoti, E. A Structure for Plans and Behavior. Elsevier, (1977) New York.

[18] Sengers, P. Do the thing right: An architecture for action-expression. In *Proceedings of the Second Int'l Conference on Autonomous Agents* (1998), 24-31.

[19] Stone, B. and Lester, J. Dynamically sequencing an animated pedagogical agent. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence* (1996), 424-431.

[20] Vilhjálmsson, H. and Cassell, J. BodyChat: Autonomous communicative behaviors in avatars. In *Proceedings of the Second International Conference on Autonomous Agents* (1998), 269-276.