Animated Pedagogical Agents for Intelligent Edutainment*

James C. Lester and **Michael A. O'Leary** and **Brian A. Stone**
Multimedia Laboratory
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206
{lester, maoleary, bastone}@eos.ncsu.edu

Abstract

One of the most important developments in the software industry in recent years has been the merger of the entertainment and educational software markets. An intriguing possibility for edutainment software is the introduction of intelligent life-like characters. By coupling inferential capabilities with well-crafted animated creatures, it becomes possible to create *animated pedagogical agents* that provide communicative functionalities with strong visual impact. We have developed the *coherence-structured behavior space* approach to creating animated pedagogical agents. This is a two-step approach. First, we design a behavior space of animation and audio segments that are structured by prerequisite relationships and a continuity metric. Second, we navigate coherent paths through the space to dynamically sequence behaviors. This creates seamless global behaviors that communicate fundamental knowledge and provide contextualized problem-solving advice. The coherence-structured behavior space approach has been implemented in "Herman the Bug," an animated pedagogical agent for DESIGN-A-PLANT, which is an edutainment system for teaching students about botanical anatomy and physiology. Formative evaluations of Herman with middle school students are encouraging.

Introduction

One of the most important developments in the software industry in recent years has been the merger of the entertainment and educational software markets. Popularly known as "edutainment," it is designed to simultaneously entertain and instruct. Because of educators' belief that well-motivated students will be more likely to exhibit the persistence required to master a complex domain, we have begun to witness an ever increasing acceptance of a technology that has traditionally been more closely associated with video games than with educational software.

Perhaps the most intriguing possibility on the horizon for edutainment is the introduction of intelligent life-like characters. By coupling inferential capabilities with well-crafted animated creatures, it becomes possible to create *animated pedagogical agents* that provide communicative functionalities with strong visual impact. These agents could observe students' progress and provide them with highly contextualized problem-solving advice in an entertaining manner. Currently, knowledge-based graphical simulations (Hollan, Hutchins, & Weitzman 1987) are virtually *de rigueur* in contemporary learning environments, the problem of planning multimedia presentations has been the subject of much study (André *et al.* 1993; Feiner & McKeown 1990; Maybury 1991; Roth, Mattis, & Mesnard 1991; Mittal *et al.* 1995), work on "self-animating" characters (Loyall & Bates 1993; Bates 1994; Blumberg & Galyean 1995; Tu & Terzopoulos 1994) is receiving increasing attention.

In this paper,[1] we propose a framework for constructing animated pedagogical agents and describe a computational mechanism for dynamically sequencing their behaviors. In this framework, creating an agent consists of constructing a behavior space, imposing a coherence structure on it, and developing a behavior sequencing engine that dynamically selects and assembles behaviors (Figure 1). This approach creates seamless global behaviors in which the agent provides visually contextualized problem-solving advice. In addition, by attending to temporal resources, it selects and composes explanatory behaviors so as to achieve the greatest coverage of the domain within the allotted time.

This framework has been used to implement an animated pedagogical agent, "Herman the Bug" (Figure 2), for DESIGN-A-PLANT (Lester *et al.* to appear), an edutainment system for botanical anatomy and physiology. Given a set of environmental conditions, children use DESIGN-A-PLANT (Lester *et al.* to appear) to graphically assemble a customized plant that can survive in the specified environment. In response to changing problem-solving contexts in DESIGN-A-PLANT, a sequencing engine orchestrates the agent's actions by selecting and assembling behaviors from a

*Support for this work was provided by the IntelliMedia Initiative of North Carolina State University and donations from Apple and IBM.

[1] An extended version of this paper appears in AAAI-96.

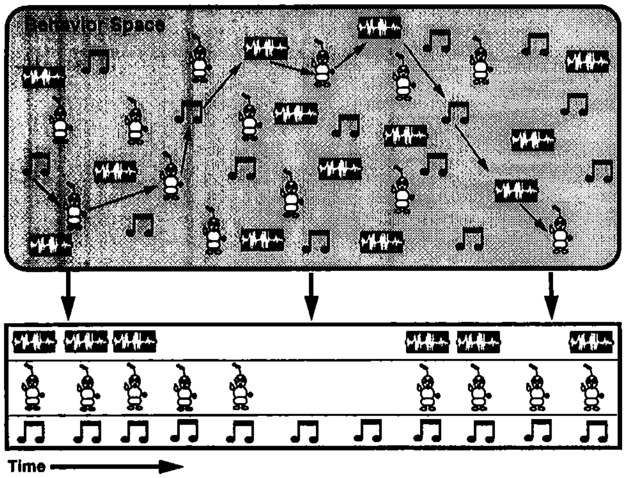

Figure 1: Sequencing coherence-structured behaviors

behavior space of 30 animations and 160 audio clips that were created by a team of 12 graphic artists and animators. It also employs a large library of runtime-mixable soundtrack elements to dynamically compose a score that complements the agent's activities. Formative evaluations of Herman conducted with middle school students are encouraging.

Designing Behavior Spaces

To provide an agent with the flexibility required to respond to a broad range of problem-solving contexts, its behavior space must be populated with a large, diverse set of animated and audio-primary behaviors. In contrast to the linear storyboarding approach employed in traditional animation (Noake 1988), the pedagogical and visual connectivity of behavior spaces require a *networked storyboarding* approach. Posing significant pedagogical and aesthetic challenges, the design of a networked storyboard is a complex, labor-intensive task requiring a multi-disciplinary team of computer scientists, graphic artists, and animators. Networked storyboarding consists of designing specifications for eight classes of animated and audio-primary behaviors and imposing a coherence structure on them.

Specifying Behaviors

Creating the agent's behavior repertoire entails setting forth precise visual and audio specifications that describe in great detail the agent's actions and utterances, rendering the actions, and creating the audio clips. The core of a behavior space is a highly interconnected web of animated segments depicting the agent performing a variety of explanatory behaviors. This is complemented by a set of audio clips of the agent's audio-primary utterances, as well as soundtrack elements (not discussed here) for the dynamically created score. To assist the sequencing engine in assembling behaviors that exhibit visual coherence, it is critical that the specifications for the animated segments take into account continuity. Accordingly, we adopt the principle of *visual bookending* to create animated segments that can more easily be assembled into visually coherent global behaviors. Visually bookended animations begin and end with frames that are identical. Just as walk cycles and looped backgrounds can be seamlessly composed, visually bookended animated behaviors can be joined in any order and the global behavior will always be flawlessly continuous.

It is important to note that visual bookending should be applied to topically-partitioned clusters of animated segments. In theory, all segments could begin and end with identical frames, but the global behaviors assembled from such a behavior space would depict the agent repeatedly leaving and returning to a single location. Because this would compromise visual coherence in most domains, partitioning the behavior space into clusters and then bookending segments within a cluster will yield superior global behaviors.

To construct a behavior space for an animated pedagogical agent, eight families of behaviors are specified collaboratively by the multi-disciplinary agent design team and then rendered by the graphic designers and animators:

- **Conceptual Explanatory Animations:** The agent explicates the structures and functions of the primary pedagogical object. For example, the DESIGN-A-PLANT agent's behavior space contains an animated segment of the agent explaining how root hairs absorb water through osmosis.
- **Problem-Solving Advisory Animations:** The agent provides abstract, principle-based advice. Students must then operationalize this advice in their problem solving activities. For example, one animated segment of the DESIGN-A-PLANT agent depicts him pointing out the relation between leaf size and low sunlight (plants in limited sunlight sometimes have larger leaves).
- **Animated Transition Segments:** These portray the agent moving from one *keyframe* (a frame initiating or terminating a segment in a bookended cluster) to another keyframe, or performing an action that will set the stage for several behaviors.
- **Audio-Primary Problem Overviews:** The agent introduces a student to a new problem. For example, the DESIGN-A-PLANT agent's behavior space contains audio clips of the agent describing environmental conditions. These utterances are played at the beginning of problem-solving episodes.
- **Audio-Primary Advisory Reminders:** The agent briefly reminds a student about principle-based advice that was presented earlier. For example, an audio clip in the DESIGN-A-PLANT agent's behavior space is a voiceover of the agent stating, "Remember that small leaves are struck by less sunlight."

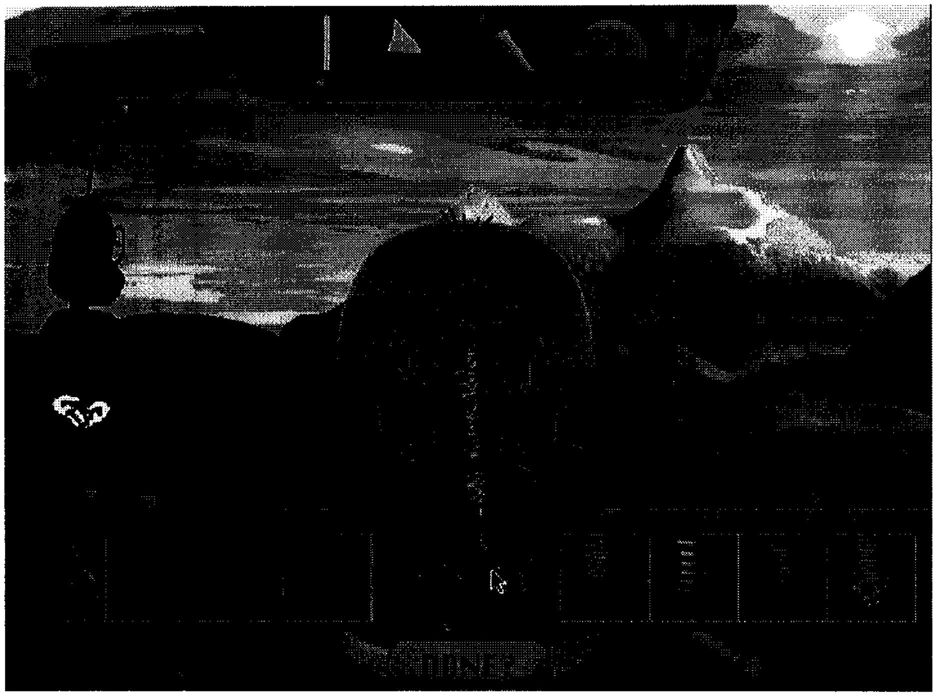

Figure 2: DESIGN-A-PLANT's animated pedagogical agent, Herman the Bug

- **Audio-Primary Direct Suggestions:** The advice presented by the agent is immediately operationalizable. For example, the DESIGN-A-PLANT agent's behavior space contains a voiceover of the agent stating, "Choose a long stem so the leaves can get plenty of sunlight in this dim environment." The agent makes these types of suggestions when a student is experiencing serious difficulties.
- **Audio-Primary Interjections:** The agent remarks about the student's progress and makes off-the-cuff comments. For example, the DESIGN-A-PLANT agent's behavior space includes Audio-Primary Interjections in which the agent congratulates the student about the successful completion of a plant design. Because a large repertoire of interjections contributes significantly to an agent's believability, a behavior space should include a variety of Audio-Primary Interjections.
- **Audio-Primary Transitions:** The agent makes meta-comments that signal an upcoming behavior. For example, the DESIGN-A-PLANT agent's Audio-Primary Transitions include a clip of him stating "It seems we're having some difficulty. Let's see if this helps ..."

Imposing a Coherence Structure

Once the behavior space has been created, it must then be structured to assist the sequencing engine in selecting and assembling behaviors that are coherent. Charting the topology of a behavior space is accomplished by constructing a tripartite behavior index, imposing a prerequisite structure on the explanatory behaviors, and creating annotations that indicate visual continuities between behaviors.

Tripartite Behavior Index. Just as the indexing of stories and advice is critical for case-based learning environments (Edelson 1993), indexing behaviors is of paramount importance for animated pedagogical agents. To enable rapid access to appropriate behaviors so they can be efficiently sequenced at runtime, behaviors are indexed ontologically, intentionally, and rhetorically. First, an *ontological* index is imposed on explanatory behaviors. Each behavior is labeled with the structure and function of the aspects of the primary pedagogical object that the agent discusses in that segment. For example, explanatory segments in the DESIGN-A-PLANT agent's behavior space are labeled by (1) the type of botanical structures discussed, e.g., anatomical structures such as roots, stems, and leaves, and by (2) the physiological functions they perform, e.g., photosynthesis. Second, an *intentional* index is imposed on advisory behaviors. Given a problem-solving goal, this structure enables the sequencing engine to identify the advisory behaviors that help the student achieve the goal. For example, one of the DESIGN-A-PLANT agent's behaviors indicates that it should be presented to a student who is experiencing difficulty with a "low water table" environment. Finally, a *rhetorical* index is imposed on audio-primary segments. This indicates the rhetorical role played by each clip, e.g., introductory remark, interjection, or congratulatory remark.

Prerequisite Structure. The primary goal of an animated pedagogical agent is to guide students through a complex subject by clearly explaining difficult concepts and offering context-sensitive problem-solving advice. To assist the sequencing engine in making decisions about the selection of behaviors, we impose a prerequisite structure on the explanatory behaviors. Prerequisite relations impose a partial order on explanatory behaviors: a behavior can be performed only if all its (immediate and indirect) prerequisite behaviors have been performed. Prerequisites should be imposed conservatively; by imposing only those relations that are clearly mandated by the domain, greater flexibility is provided to the sequencing engine because the number of behaviors it may select at any given time will be greater.

Visual Continuity Annotations. Because visual bookending is not always possible, the behavior space should include knowledge about the visual continuities between animated segments in the prerequisite structure. Visual attributes including the shot's zoom level and the agent's frame position are represented as normalized numerical variables and are assigned weights based on priority. The visual continuity $v_{x,y}$ between behaviors B_x and B_y is defined as the distance in n-dimensional attribute space between the final frame of B_x and the initial frame of B_y:

$$v_{x,y} = \sqrt{w_1(x_1 - y_1)^2 + w_2(x_2 - y_2)^2 + \ldots w_n(x_n - y_n)^2}$$

where w_i is the prioritized weight of the ith visual attribute. The sequencing engine uses the continuity annotations to maximize visual continuity among sequenced animated segments.

Sequencing Agents' Behaviors

To achieve agent persistence, agent immersion, and pedagogical object persistence, the agent remains on-screen, visually immersed in the learning environment, and on or near the primary pedagogical object at all times. The moment a student requests assistance, constructs an incorrect (complete or partial) solution, or fails to take action for an extended period of time, the sequencing engine (Figure 3) is called into play to create the agent's next behavior. By exploiting the behavior space's coherence structure and noting different aspects of the current problem-solving context, the sequencing engine navigates through the space to weave the local behaviors into global behaviors. It employs the following algorithm to select and assemble local behaviors in real time:

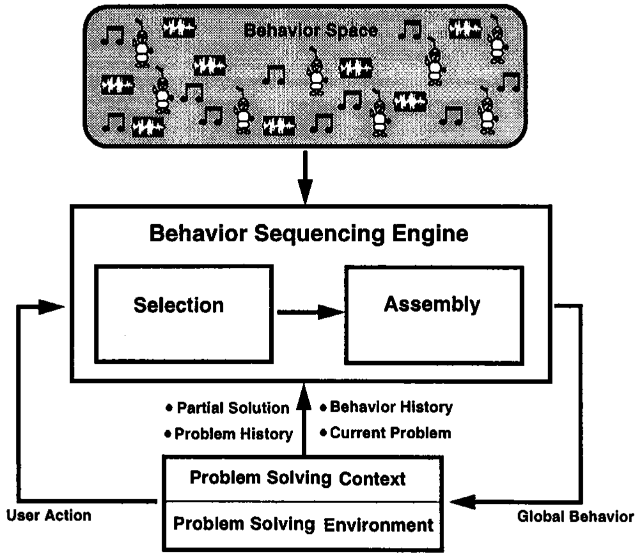

Figure 3: The behavior sequencing engine

1. **Compute n, the number of explanatory behaviors to exhibit.** This quantity is computed by $\lfloor b/f \rfloor$. The quantity b is the number of explanatory behaviors that have not yet been exhibited. The function f, which is determined from empirical data, is the predicted number of future problem-solving situations in which explanatory behaviors can be exhibited.[2] The floor is taken for non-integer results to be conservative—representing the number of Conceptual Explanatory Animated Segments that should be exhibited. Employing n has the effect of evenly distributing these explanations over the course of the learning session.
2. **Select all explanatory behaviors E^P that are pedagogically viable.** First, apply the ontological index structure to index into behavior space and identify all Conceptual Explanatory Animated Segments that are currently relevant. By noting the current structures, functions, and problem-solving features that are active in the current problem, the sequencing engine can identify the animations that are pedagogically appropriate. Second, determine candidate behaviors whose prerequisite behaviors have already been exhibited by using the prerequisite structure to perform a topological sort of behaviors in the global behavior history.
3. **Select explanatory behaviors $E^{P,V}$ that are both pedagogically and visually viable.** Of the candidates in E^P chosen in Step 2, select a subset $E^{P,V}$ such that (a) the sum of the continuity annotations along the best path in $E^{P,V}$ is minimized, and (b) $\mid E^{P,V} \mid$ is as close as possible to n without exceeding it.[3]

[2]For example, DESIGN-A-PLANT has sixteen problems segregated into four complexity levels. We have collected empirical data with students using the system to determine f for each problem.

[3]Note that Steps (2) and (3) must be interleaved when selecting multiple behaviors because prerequisites can be

4. **Select problem-solving advisory behaviors A that are pedagogically appropriate.** Use the intentional and rhetorical indices to identify advisory behaviors that are germane to the topic of the current problem. A may include both animated and audio-primary behaviors.
5. **Select the media with which to exhibit a subset A' of the behaviors in A.** Inspect the behavior history to determine if advisory behaviors about the current topic have been exhibited. If no prior advisory behaviors on this topic have been presented, select an animated advisory behavior on this topic. If an animated advisory behavior on this topic has been previously exhibited, select an audio-primary verbal reminder on this topic. If an animated advisory behavior on this topic has been previously exhibited but a significant amount of time has elapsed, select it for repeat viewing. If both an animated advisory behavior and a verbal reminder on this topic have been exhibited recently, select an audio-primary direct behavior in which the agent will explicitly tell the student what problem-solving action to take.
6. **Select animated and verbal transitions T.** Use the indices and prerequisite structure to identify transition behaviors for $E^{P,V}$ and A'.
7. **Assemble the final global behavior.** Impose the following temporal ordering on the selected behaviors: (a) verbal transitions in T to introduce the upcoming explanations; (b) animated explanatory behaviors in $E^{P,V}$ ordered by prerequisite structure; (c) animated advisory behaviors in A'; and (d) audio-primary reminders and direct advisory behaviors in A'.

The resulting global behavior is presented onscreen and the sequencing engine sleeps until the next invocation. While it is sleeping, it pseudo-randomly schedules audio-primary interjections. In addition, the agent's actions are complemented at all times by a continuous soundtrack whose voicing and tempo are dynamically updated to reflect changes in problem-solving contexts. Introductory measures are played as problems are introduced, and additional voicing is added as partial solutions are successfully constructed. The net effect of the sequencing engine's activities is students' perception that a life-like character is carefully observing their problem-solving activities and moving in and out of the primary pedagogical object to provide advice just when it is needed.

An Implemented Animated Agent

The coherence-based approach to dynamic sequencing has been implemented in Herman the Bug, an animated pedagogical agent for DESIGN-A-PLANT, which is a learning environment being developed in our laboratory to teach middle school students about botanical anatomy and physiology.[4] Herman is a talkative, quirky, somewhat churlish insect with a propensity to fly about the screen and dive into the plant's structures as it provides students with problem-solving advice. His behavior space consists of 30 animated segments[5] (twenty are in the 20—30 second range and ten are in the 1–2 minute range), 160 audio clips, several songs, and a large library of runtime-mixable, soundtrack elements. Throughout the learning session, he remains onscreen, standing on the plant assembly device when he is inactive (Figure 2) and diving into the plant as he delivers advice visually. In the process of explaining concepts, he performs a broad range of activities including walking, flying, shrinking, expanding, swimming, fishing, bungee jumping, teleporting, and acrobatics. All of his behaviors are sequenced in real time on a Power Macintosh 9500/132.

To gauge the effectiveness of the coherence-based approach to dynamically sequencing the behaviors of animated pedagogical agents, formative observational studies were conducted with thirteen middle school students using Herman with DESIGN-A-PLANT. Each student interacted with the learning environment for forty-five minutes to one hour. As the students designed plants for a variety of environmental conditions, Herman introduced problems, explained concepts in botanical anatomy and physiology, provided problem-solving advice, and interjected congratulatory and off-the-cuff remarks. These studies suggest that animated pedagogical agents whose behaviors are selected and assembled with the sequencing engine can effectively guide students through a complex subject in a manner that exhibits both pedagogical and visual coherence.

Herman was unanimously well received. Its pedagogical and visual coherence, together with its immersive property—the fact that it inhabits a 3D environment and interacts with 3D plant models to explain structural and functional concepts—produced strikingly life-like behaviors. Herman's visual behaviors seemed to flow so well that no student commented or displayed surprise during transitions. Because of bookending, many of his transitions were technically flawless. Herman's verbal reminders enabled students

met dynamically in the process of exhibiting a global behavior.

[4] DESIGN-A-PLANT is a *design-centered* learning environment that embodies a strong constructivist approach to learning. Students use it to graphically assemble customized 3D plants from a library of plant anatomical structures. Their goal in each design episode is to create a plant that will survive under a specific set of environmental conditions. At the implementational level, DESIGN-A-PLANT is a constraint-based system, where the constraints imposed by the plant's environment must be satisfied by the anatomical structures selected by the student.

[5] Its animations were designed, modeled, and rendered on SGIs and Macintoshes by twelve graphic artists and animators.

to continue with their problem solving uninterrupted, and during the study students made frequent (and unprompted) positive comments about the agents physical actions and remarks. The variety of his behaviors maintained their interest throughout the session, and every student, without exception, commented positively about the continuously updated score. Perhaps not surprisingly—considering its seventh grade audience—Herman's quirky asides were well received.

Conclusion

Animated pedagogical agents can combine adaptive explanatory behaviors with great visual appeal. We have proposed an approach to dynamically sequencing agents' behaviors that exploits (1) a behavior space containing animated and verbal behaviors, and (2) a coherence structure consisting of a tripartite behavior index of ontological, intentional, and rhetorical indices, a prerequisite structure, and continuity annotations that estimate the degree of visual continuity between pairs of behaviors. By navigating the behavior space and attending to the coherence structure, a behavior sequencing engine selects and assembles behaviors that exhibit both pedagogical and visual coherence. This coherence-based approach to behavior sequencing has been implemented in an agent that operates in real time to dynamically sequence behaviors in response to rapidly changing problem-solving contexts. It has been tested in a learning environment with middle school children, and the results are encouraging. We are currently evaluating five different versions of Herman with 100 middle school students.

This work represents a promising first step toward creating animated pedagogical agents for edutainment systems. The potential scope of their application is quite broad, ranging from K-12 and higher education to corporate training. Though Herman now has a relatively large repertoire of behaviors, perhaps the greatest challenge lies in increasing his flexibility. A promising technique for accomplishing this is to reduce the granularity of his behaviors, an approach we are currently exploring.

Acknowledgements

Thanks to: the animation team which was lead by Patrick FitzGerald; the students in the Intelligent Multimedia Communication, Multimedia Interface Design, and Knowledge-Based Multimedia Learning Environments seminars; Chris Tomasson and her seventh grade class at Martin Middle School for participating in the evaluation; and Patrick FitzGerald and Charles Callaway for comments on an earlier draft of this paper.

References

André, E.; Finkler, W.; Graph, W.; Rist, T.; Schauder, A.; and Wahlster, W. 1993. WIP: The automatic synthesis of multi-modal presentations. In Maybury, M. T., ed., *Intelligent Multimedia Interfaces.* AAAI Press. chapter 3.

Bates, J. 1994. The role of emotion in believable agents. *Communications of the ACM* 37(7):122–125.

Blumberg, B., and Galyean, T. 1995. Multi-level direction of autonomous creatures for real-time virtual environments. In *Computer Graphics Proceedings*, 47–54.

Edelson, D. C. 1993. *Learning from Stories: Indexing and Reminding in a Socratic Case-Based Teaching System for Elementary School Biology.* Ph.D. Dissertation, Northwestern Univeristy.

Feiner, S. K., and McKeown, K. R. 1990. Coordinating text and graphics in explanation generation. In *Proceedings of the Eighth National Conference on Artificial Intelligence*, 442–449.

Hollan, J. D.; Hutchins, E. L.; and Weitzman, L. M. 1987. STEAMER: An interactive, inspectable, simulation-based training system. In Kearsley, G., ed., *Artificial Intelligence and Instruction: Applications and Methods.* Reading, MA: Addison-Wesley. 113–134.

Lester, J.; Stone, B.; O'Leary, M.; and Stevenson, R. to appear. Focusing problem solving in design-centered learning environments. In *Proceedings of the Third International Conference on Intelligent Tutoring Systems.*

Loyall, A. B., and Bates, J. 1993. Real-time control of animated broad agents. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society.*

Maybury, M. T. 1991. Planning multimedia explanations using communicative acts. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, 61–66.

Mittal, V.; Roth, S.; Moore, J. D.; Mattis, J.; and Carenini, G. 1995. Generating explanatory captions for information graphics. In *Proceedings of the International Joint Conference on Artificial Intelligence.*

Noake, R. 1988. *Animation Techniques.* London: Chartwell.

Roth, S. F.; Mattis, J.; and Mesnard, X. 1991. Graphics and natural language as components of automatic explanation. In Sullivan, J. W., and Tyler, S. W., eds., *Intelligent User Interfaces.* New York: Addison-Wesley. 207–239.

Tu, X., and Terzopoulos, D. 1994. Artificial fishes: Physics, locomotion, perception, and behavior. In *Computer Graphics Proceedings*, 43–50.