Lifelike Pedagogical Agents for Mixed-Initiative Problem Solving in Constructivist Learning Environments

JAMES C. LESTER,* BRIAN A. STONE ** and GARY D. STELLING ‡
Department of Computer Science
North Carolina State University

Abstract. Mixed-initiative problem solving lies at the heart of knowledge-based learning environments. While learners are actively engaged in problem-solving activities, learning environments should monitor their progress and provide them with feedback in a manner that contributes to achieving the twin goals of learning effectiveness and learning efficiency. Mixed-initiative interactions are particularly critical for *constructivist* learning environments in which learners participate in active problem solving. We have recently begun to see the emergence of believable agents with lifelike qualities. Featured prominently in constructivist learning environments, *lifelike pedagogical agents* could couple key feedback functionalities with a strong visual presence by observing learners' progress and providing them with visually contextualized advice during mixed-initiative problem solving. For the past three years, we have been engaged in a large-scale research program on lifelike pedagogical agents and their role in constructivist learning environments. In the resulting computational framework, lifelike pedagogical agents are specified by (1) a *behavior space* containing animated and vocal behaviors, (2) a *design-centered context model* that maintains constructivist problem representations, multimodal advisory contexts, and evolving problem-solving tasks, and (3) a *behavior sequencing engine* that in realtime dynamically selects and assembles agents' actions to create pedagogically effective, lifelike behaviors. To empirically investigate this framework, it has been instantiated in a full-scale implementation of a lifelike pedagogical agent for DESIGN-A-PLANT, a learning environment developed for the domain of botanical anatomy and physiology for middle school students. Experience with focus group studies conducted with middle school students interacting with the implemented agent suggests that lifelike pedagogical agents hold much promise for mixed-initiative learning.

Key words: Lifelike agents, pedagogical agents, animated agents, knowledge-based learning environments, mixed-initiative interaction, intelligent tutoring systems, intelligent multimedia presentation, intelligent interfaces, task models.

1. Introduction

Mixed-initiative problem solving lies at the heart of knowledge-based learning environments. Since the birth of the field more than twenty-five years ago (Car-

* Department of Computer Science, Engineering Graduate Research Center, North Carolina State University, Raleigh, NC 27695-7534, U.S.A. Email: lester@csc.ncsu.edu
** Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8206, U.S.A. Email: bastone@eos.ncsu.edu
‡ Department of Computer Science, Engineering Graduate Research Center, North Carolina State University, Raleigh, NC 27695-7534, U.S.A. Email: gdstelli@eos.ncsu.edu

bonell, 1970), it has become apparent that developing computational models of mixed-initiativity is critical to the learning environment enterprise. While learners are actively engaged in problem solving activities, learning environments should monitor their progress and provide them with feedback in a manner that contributes to achieving the twin goals of learning effectiveness and learning efficiency. By carefully monitoring a learner's progress, learning environments should control the course of the interaction in such a way that they maximize the quality of the learning experience.

We have recently begun to see the emergence of believable agents with lifelike qualities (Bates, 1994; Blumberg and Galyean, 1995; Kurlander and Ling, 1995; Maes et al., 1995; André and Rist, 1996). By building on developments in these intriguing interactive characters, we can create a new generation of knowledge-based learning environments that are inhabited by animated *lifelike pedagogical agents*. Featured prominently in learning environments, they could couple key feedback functionalities with a strong visual presence by observing learners' progress and providing them with visually contextualized problem-solving advice.

Lifelike pedagogical agents offer particularly significant potential for *constructivist* learning environments. Constructivist learning (Piaget, 1954) has received increasing attention in the education community in recent years because of its emphasis on the active role played by the learner as he or she acquires new concepts and procedures. A particularly intriguing form of the constructivist's learning-by-doing techniques is "learning-by-designing." In the process of designing an artifact, learners—by necessity—come to understand the rich interconnections between the artifacts they devise and the environmental constraints that determine whether a given design will meet with success. Because design tasks are inherently complex, design-centered problem solving provides an excellent testbed for studying mixed-initiative interactions that are contextualized in a learners' problem-solving activities.

To investigate these issues, we have been engaged in a large-scale research program on lifelike pedagogical agents and constructivist learning environments (Lester et al., 1996; Stone and Lester, 1996; Lester and Stone, 1997; Lester et al., 1997a; Lester et al., 1997b; Lester et al., 1997c). The long-term goal of the project is to create pedagogically effective computational mechanisms that contribute to fundamental improvements in learning environments. To date, we have focused on developing a *pedagogical agent behavior sequencing engine* that dynamically controls the behaviors of lifelike pedagogical agents in response to the rapidly changing problem-solving contexts in constructivist learning environments. Applying this framework to create an agent entails constructing a behavior space, a design-centered context model, and a behavior sequencing engine that dynamically selects and assembles behaviors:

1. **Agent Behavior Space:** A behavior space contains (a) animated behaviors of an agent performing a variety of pedagogical behaviors including explanatory and advisory actions, (b) animated behaviors of the agent engaged in a variety

of "believability-enhancing" actions, and (c) narrative utterances spoken by the agent, including verbal reminders and interjections.

2. **Design-Centered Context Model:** A model of a design-centered context consists of (a) an *environmental context* representing the critical features of the problem, (b) a *multimodal advisory history* representing the explanations and advice that have been presented by the agent, and (c) an *artifact-based task model* representing the features of the artifact being designed by the learner. These are dynamically updated as problem-solving episodes unfold.
3. **Behavior Sequencing Engine:** At runtime, a behavior sequencing engine orchestrates an agent's behaviors in response to the changing problem-solving context by exploiting the design-centered context model. A sequencing engine selects an agent's actions by navigating coherent paths through the behavior space and assembling them dynamically to create global behaviors in which the agent provides visually contextualized problem-solving advice.

To empirically investigate this framework, it has been instantiated in an implemented lifelike pedagogical agent, Herman the Bug (Figure 1), who inhabits a constructivist learning environment for the domain of botanical anatomy and physiology for middle school students, DESIGN-A-PLANT.* The agent interactively provides contextualized advice to learners as they graphically assemble plants from a library of plant structures such as roots and stems.

In DESIGN-A-PLANT, a learner's goal in each problem-solving episode is to design a plant that will thrive in a given natural environment with specified conditions such as the amount of available sunlight. As learners solve problems by constructing plants, the agent provides them with advice about botanical anatomy and physiology. Together, Herman and the DESIGN-A-PLANT learning environment constitute a proof-of-concept embodiment of the lifelike pedagogical agent behavior sequencing framework and provide a "laboratory" for studying mixed-initiative problem-solving interactions in constructivist learning environments.

Based on the experience of the first three years of the DESIGN-A-PLANT project, this article provides an account of the representations and computational mechanisms underlying the design and construction of lifelike pedagogical agents for mixed-initiative problem solving in constructivist learning environments. It is structured as follows. Section 2 sets forth design criteria for mixed-initiativity in lifelike pedagogical agents, describes the DESIGN-A-PLANT learning environment testbed, and presents an extended mixed-initiative session to illustrate the desired phenomena. Section 3 presents the design-centered context model used to represent problem-solving and advisory contexts. Section 4 describes dynamic behavior sequencing engines for lifelike agents, including the mechanisms for controlling the initiative, for making intervention decisions, and for interjecting advice

* The DESIGN-A-PLANT learning environment project is a large-scale multidisciplinary project involving computer scientists, animators, graphic designers, voice actors, curriculum and instruction specialists, and cognitive scientists. For example, all of the 3D graphics and animations were designed, modeled, and rendered by a twelve-person graphic design and animation team.

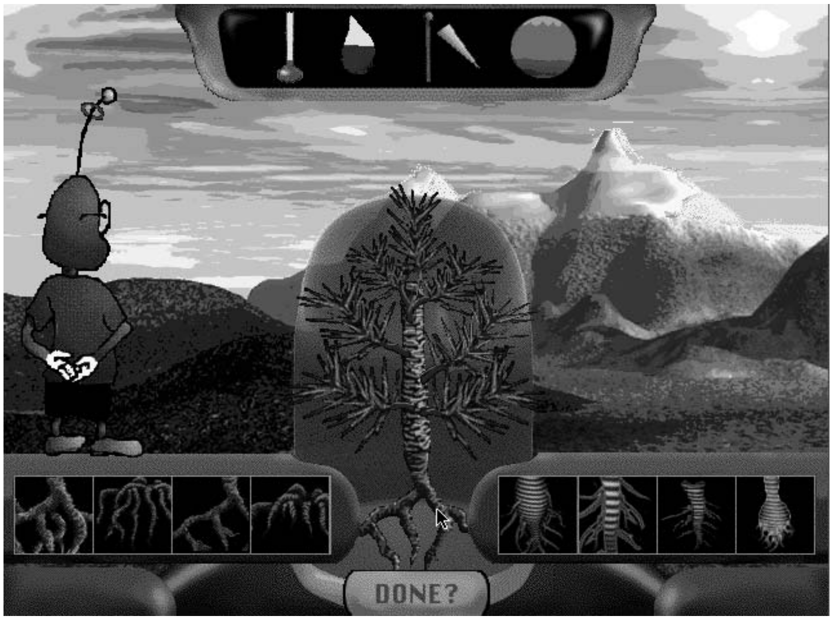

Figure 1. The DESIGN-A-PLANT lifelike pedagogical agent

and explanations. Section 5 illustrates these computational mechanisms with an extended example interaction. Section 6 puts the work in perspective by describing the principle "lessons learned" from the experiences of iterative design, implementation, and evaluation via focus group studies with middle school students. Section 7 concludes with a summary and a discussion of future directions.

2. Mixed-Initiativity in Lifelike Pedagogical Agents

Since their conception more than a quarter of a century ago, knowledge-based learning environments (Hollan et al., 1987; Wenger, 1987; Lesgold et al., 1992; Anderson et al., 1995) have offered significant potential for fundamentally changing the educational process. It has long been believed—and recently rigorously demonstrated (Mark and Greer, 1995)—that presenting knowledgeable feedback to students increases learning effectiveness. Despite this promise, few learning environments have made the difficult transition from the laboratory to the classroom, and the challenge of developing learning environments that are both pedagogically sound and visually appealing has played no small part in this situation.

Lifelike animated agents could play a central communicative role in learning environments by providing visually contextualized problem-solving advice. Although knowledge-based graphical simulations (Hollan et al., 1987) are virtually *de rigueur* in contemporary learning environments, and the problem of planning multimedia presentations has been the subject of much study (André et al., 1993;

Feiner and McKeown, 1990; Maybury, 1991; Roth et al., 1991; Mittal et al., 1995), work on lifelike agents has begun in earnest but is still in its infancy (Bates, 1994; Blumberg and Galyean, 1995; Kurlander and Ling, 1995; Maes et al., 1995; André and Rist, 1996). Despite the promise of lifelike pedagogical agents, with the exception of work on the DESIGN-A-PLANT project (Lester et al., 1996; Stone and Lester, 1996; Lester and Stone, 1997; Lester and Stone, 1997; Lester et al., 1997b; Lester et al., 1997c) (described in this article) and the Soar Training Expert for Virtual Environments (STEVE) project (Rickel and Johnson, 1997), which focuses on agents that provide instruction about procedural tasks, lifelike agents for pedagogy have received little attention.

In the same manner that human-human tutorial dialogues are characterized by changes in initiative (Smith and Hipp, 1994; Hale and Barsalou, 1995; Freedman, 1996), learner-agent interactions between a lifelike agent and a learner should be characterized by problem-solving episodes where control of the initiative frequently changes. Mixed-initiativity is such an extraordinarily complex phenomenon (Cohen et al., 1998), that developing a computational model of mixed-initiative tutorial interactions is especially challenging. Below we characterize the kinds of initiative changes we target for tutorial interactions.

At the beginning of each episode, learners are unfamiliar with the problem, so the agent should take control and introduce the problem. For example, in the DESIGN-A-PLANT learning environment, the agent should open by describing the environmental conditions that hold on the particular environment for which a plant will be designed. Once learners begin solving problems, the initiative may change frequently. Learners should be able to take control while they are performing problem-solving actions, agents should regain control when it appears that learners are experiencing difficulty or when learners request assistance, and control should then be relinquished to learners so they may continue their problem solving. For example, in the DESIGN-A-PLANT learning environment, the agent should monitor students as they assemble plants and intervene to provide explanations about botanical anatomy and physiology when they reach an impasse. In the same manner that human interlocutors engaged in mixed-initiative interactions frequently generate responses that include highly relevant information that was not specifically requested (Green and Carberry, 1998), the agent should be prepared to provide learners with assistance even though it may not be explicitly requested.

Once problem solving is successfully completed by the learner, the agent should again regain control to complete the problem-solving transaction. This might involve a simple statement that a correct solution has been constructed or perhaps a more elaborate congratulatory utterance accompanied by a visually compelling behavior. For example, at the end of each successful problem-solving episode in DESIGN-A-PLANT, the agent might congratulate learners and cartwheel across the screen.

Well designed intervention strategies are especially critical in constructivist learning environments. The frequency and content of intervention should be appropriate for the particular aspects of the design task on which the learner is focusing, and advice should be relevant to the problem-solving goals currently being pursued. In the same manner that coherence plays a critical role in assisting readers' comprehension of text (Grimes, 1975), the behaviors of an animated pedagogical agent should be molded by considerations of pedagogical coherence.

Perhaps most central among these requirements is that an agent's advisory and explanatory interjections be *situated* (Brown et al., 1989): all of its explanatory behaviors—not merely its advisory actions but also its communication of fundamental conceptual knowledge—should take place in concrete problem-solving contexts. For example, learners interacting with the DESIGN-A-PLANT environment should learn about leaf morphology in the context of selecting a particular type of leaf as they design a plant that will thrive in particular environmental conditions. Moreover, agents' behaviors should obey prerequisite relationships and include transitions (both verbal and visual) that are the hallmark of spoken discourse.

Creating agents that intervene appropriately requires inferring the learner's intentions. However, diagnosis should be conducted as non-invasively as possible because continually interrupting learners to determine their current intent and to ferret out their misconceptions would interrupt constructivist learning. For example, it would go against the spirit of constructivist learning to prevent the learner from pursuing his or her design activities in order to issue a number of probes to detect precisely which misconceptions were active at a given time.

To achieve mixed-initiative interaction with lifelike agents, believability (Bates, 1994) is a key feature of these agents. We define the *believability* of lifelike agents as the extent to which users interacting with them come to believe that they are observing a sentient being with its own beliefs, desires, intentions, and personality. Although it is possible that increasing believability may yield substantial rewards in learners' motivation, when learners have the initiative, agents must exhibit believability-enhancing behaviors such as standing up and sitting down in such a manner that they do not distract from problem solving. Some behaviors such as moving from one location to another have high visual impact, while others, such as small head movements, have low visual impact. In general, the higher the visual impact, the more interesting a behavior will be, but agents must control the visual impact of their behaviors in such a manner that they do not divert a learner's attention at critical junctures.

In short, agents' behaviors should be sequenced in such a way that, in the normal course of a learner's problem-solving activities, the transfer of initiative between the learner and the agent plays out as smoothly as possible, interventions are provided in a timely and topical manner, diagnosis is non-invasive, and believability-enhancing behaviors do not interfere with but rather enhance the learning experience.

2.1. A Testbed Learning Environment for Mixed-Initiativity

To empirically investigate mixed-initiative interaction with lifelike pedagogical agents, we have implemented an animated agent, Herman the Bug (Figure 1), who interacts with learners solving problems in the DESIGN-A-PLANT learning environment. Herman and DESIGN-A-PLANT are the central computational artifacts of a long-term project.* The implemented behavior sequencing engine operates in realtime to dynamically monitor and update the environmental context, advisory history, and task model and to select and compose the agent's behaviors approximately once every 200 milliseconds.**

The agent is a talkative, quirky insect with a propensity to fly about the screen and dive into the plant's structures as he provides problem-solving advice. 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. His *behavior space* was designed to "stress-test" the behavior sequencing algorithms and representations. Containing more than 50 animated behaviors and approximately 100 verbal behaviors, the behavior space houses a variety of pedagogical and believability-enhancing behaviors. The pedagogical behavior space includes a variety of advisory and explanatory behaviors pertaining to botanical anatomy, physiology, environmental constraints, and their interactions. The believability-enhancing behavior space includes re-orientation behaviors (e.g., standing up, lying down), restrictive body behaviors (e.g., back scratching, head scratching, toe tapping, body shifting), prop-based behaviors (e.g., glasses cleaning), and full-screen celebratory behaviors (e.g., bungee jumping, cartwheeling).

Learners interact with the agent as they graphically assemble customized 3D plants (pre-rendered on an SGI) 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. Each environment (visualized as a different imaginary planet) is rendered as an intriguing landscape. Specific environmental factors are depicted iconically, and the roots, stems, and leaves in the artifact component

* The first three years of the project have been devoted to iteratively designing and building the animated agent and its "laboratory" (the DESIGN-A-PLANT ENVIRONMENT for studying mixed-initiative human-agent interactions). The second three years will be devoted to empirical investigations of the cognitive processes and the results of constructivist human-agent learning.

** The behavior sequencing engine runs on a Power Macintosh 9500/132.

* A general goal of the DESIGN-A-PLANT project is creating learning episodes that revolve around learners' active design-centered problem solving. Although constructivism has come to dominate contemporary theories of learning, arriving at a clear definition of it has proved challenging indeed. Two closely related approaches to learning are that of constructionism and constructive learning. *Constructionist* theories emphasize that learning "happens especially felicitously in a context where the learner is consciously engaged in constructing a public entity, whether it's a sand castle on the beach or a theory of the universe" (Papert, 1991). *Constructive* learning, unlike the above, places no emphasis on authenticity or the social environment in which learning plays out. Hence, though we refer to learning in DESIGN-A-PLANT as "constructionist," we acknowledge the important differences between constructivism, constructionism, and constructive learning and intend no philosophical claims.

library are 3D objects. “Rollover” definitions are provided for all environmental factors and components. The DESIGN-A-PLANT testbed is a fully functional learning environment containing the following elements:

– *Environments:* 16 environments (4 types of environments, each with 4 complexity levels)
– *Artifact component library:* 8 types of roots, 8 types of stems, and 8 types of leaves.
– *Domain model:* 31 constraint packets that relate 6 environmental factors to the anatomical structures and physiological functions.

All interactions between learners and the agent take place in DESIGN-A-PLANT’s *design studio*. To make progress on issues of mixed-initiativity without awaiting solutions to the natural language understanding problem, the DESIGN-A-PLANT learning environment operates without an NLU component, just as the collaborative interface agent in the COLLAGEN system does (Rich and Sidner, 1998). The design studio is an interactive workbench (Figure 1) that was crafted to enable learners to attack design problems flexibly: they have the freedom to begin working on a sub-task, effortlessly move to new sub-tasks, revise design decisions in light of the agent’s advice, and return to previously considered sub-tasks with ease. To simultaneously achieve this flexibility and to enable the system to monitor their tasks non-invasively, the interface state and its functionalities are tightly coupled to the task model.

Learners design plants in the design studio’s “plant bubble.” They graphically assemble their designs by first positioning the component task bar vertically on the screen. This requires only a single mouse click. When the component task bar is in the bottom-most position, the root library is displayed; when it is mid-level, stems are displayed; and when it is at the top, leaves are displayed. Learners then indicate design decisions by choosing a component of the selected type. This also is accomplished by a single mouse click. Because all workbench actions are directly mapped to their corresponding sub-tasks, learners (perhaps unknowingly) signal their intent to the behavior sequencing engine during the natural course of the design process. When they believe their design is complete and correct, they click on the *Done* button at the bottom of the screen, and the system evaluates their plant with respect to the given environment by searching for violations of constraints in the underlying domain model.

2.2. SAMPLE MIXED-INITIATIVE TUTORIAL INTERACTION

To illustrate the nature of the desired mixed-initiative interactions we seek, consider the following series of exchanges in a DESIGN-A-PLANT learning session. A learner has watched Herman’s overview of elementary plant anatomy and has visited two planets. The first had a simple, high rainfall, environment which required her to choose thin leaves, for flexibility. In the second environment, a planet with dim sunlight and a low watertable, she needed assistance twice. She has now been

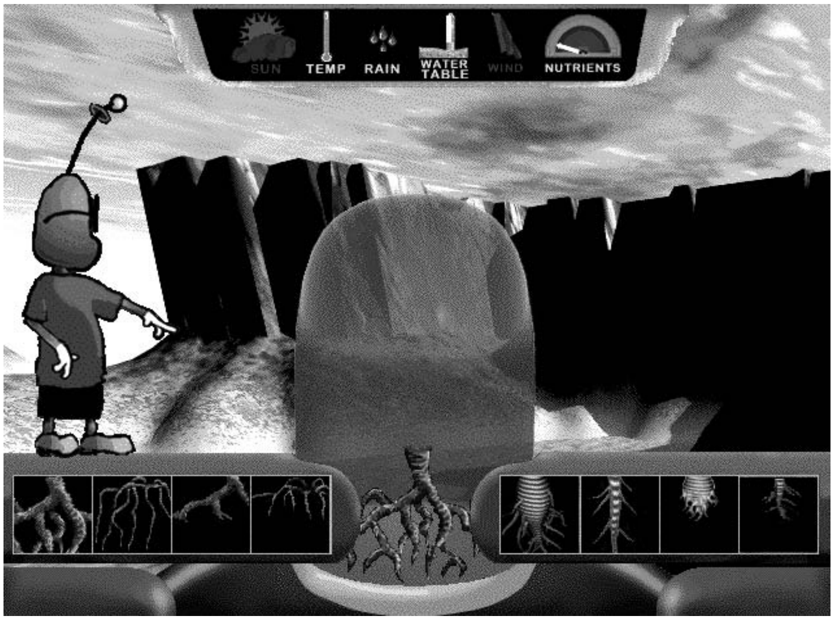

Figure 2. The learner visits a new environment and undertakes root design

escorted to a planet with low rainfall and high temperature (Figure 2).* In this environment, roots and leaves are both in the environmental focus.

The agent first takes the initiative to introduce the learner to the problem.

Animated Agent: Whoa! I'm feeling hot, hot, hot! Too bad there's no raindrops to fall upon my head. Well, the roots better be well chosen to soak up all the water they can. The stem and leaves still need to store as much water as possible, but at these high temperatures, they also need to be able to use some of that water to stay cool, by transpiration.

Initiative is then transferred from agent to the learner who begins her problem solving. In general, learners can begin with any sub-task they wish.

Learner: Opts to begin plant construction with roots.

To avoid distracting the learner while she addresses difficult problems, the agent stands quietly, attentively looking at the transparent chamber in which the learner is designing her plant.

---

* To emphasize artifacts and environmental variables that exercise concepts with which the learner is experiencing difficulty, some versions of DESIGN-A-PLANT dynamically select environments. By inspecting the task model, they not only present types of problems that have proved difficult for the learner in the past, but they also control the level of complexity (as measured by the number of constraints) that the environment will exhibit. To do so, they exploit an *environment matrix*, where each element of the matrix is an environment (Lester et al., 1997c). Each column represents a particular environmental "intent," i.e., a particular sub-task to be exercised, and each row represents additional complexity. By navigating the environment matrix, these versions of DESIGN-A-PLANT select environments that produce customized, challenging design experiences.

Learner: Spends a while considering the rollover descriptions of the environment elements and roots settings icons, but cannot make a decision.

> Because the learner has made limited progress, the agent takes the initiative and provides verbal advice about the relevant features of the component on which the learner currently focuses.

Animated Agent: Because of the light rain in this environment, one of the choices you have to make is between branching and non-branching roots. Which type would increase the roots' contact with the scarce moisture?

Animated Agent: After a slight pause, raises a similar question about deep and shallow roots.

> After offering the advice, the agent returns the initiative to the learner.

Learner: Speculating that branching roots are more efficient and, deciding that shallow roots will, in Herman's words, "gather up the moisture as it soaks into the ground," chooses roots that are branching and shallow.

> Because the learner's design decision will yield an artifact that will flourish in the given environment (assuming all of the remaining design decisions are reasonable), the agent takes the initiative to issue a verbal congratulation for the intermediate success.

Animated Agent: Congratulates the learner.

Learner: Raises the task bar to choose a stem.

Learner: Mindful of Herman's comment about storing water in the stem and leaves, chooses a thick stem that is also short and covered with bark.

> Although some aspects of the learner's design decision are reasonable, others are not. The agent intervenes to offer verbal advice about the particular aspects on which she requires assistance.

Animated Agent: Whoo-eee! If the plant's in a hot area, there's a danger it could overheat. Like me, I'm so hot! To avoid this, the plant transpires, losing water through the stomata. Which will allow the plant to transpire more freely, a stem with bark on it or a stem with no bark?

> Because the learner has demonstrated that she has understood some of the critical features of the current sub-task, the agent can now be more visually active. Even so, he waits a respectable amount of time before performing an interesting behavior, such as polishing his glasses or deploying a yo-yo, to remind the learner of his presence. The learner retains the initiative.

Learner: Correctly concludes that a stem with no bark will "transpire more freely."

> The agent takes the initiative to issue congratulations on this successfully completed sub-task and then returns the initiative to the learner.

Animated Agent: Expresses his congratulations.

Learner: Shifts the focus to the leaves sub-task.

When the learner was working on the preceding two tasks, Herman remained standing on the task bar. However, because the task bar is now high on the screen as the learner shifts to the leaves sub-task, Herman sits down as the taskbar rises so as not to bump his head on the top of the screen.

Learner: Knows to choose thick leaves for their ability to store the scarce moisture, but has not yet generalized her understanding of maximizing cooling by transpiration to choosing "thin-skinned" leaves; selects thick, large, thick-skinned leaves.

Because one of the features of the component currently under consideration will result in an artifact that cannot satisfy the current environmental constraints (the learner has made a design decision that is valid with respect to the leaves' overall thickness feature but not with respect to their skin thickness), the agent will intervene to provide advice about the specific problematic features.

Animated Agent: In a hot climate, a plant needs to release excess water as vapor to cool itself off. Huh-huh. Basically it's sweating, just as we do, to prevent itself from overheating. Choose between those leaves which have a thick skin and those with a thin skin; which is better to keep the plant from overheating?

The agent returns the initiative to the learner.

Learner: Makes a second incorrect design decision of thick, small leaves with thick skin.

The learner has again demonstrated that she may have a misconception about the relationship between skin thickness and the active environmental attributes. The agent therefore intervenes. He first takes advantage of the opportunity to provide an animated explanation about external leaf anatomy. Then, because it appears that the learner was unable to operationalize the abstract verbal advice provided above, he visually advises the learner about the anatomical requirements imposed by this particular environment. The intervention begins with a verbal meta-comment about the upcoming interjection, after which the agent lies down on the task bar prior to presenting the explanations and advice. At their conclusion he returns to his original orientation.

Animated Agent: OK, OK, so we're having some difficulty. But, that's OK, we're here to learn. I tell you what, see if this helps.

Animated Agent: Stretches out on the task bar to watch the animations ("home movies" of himself interacting with the plant) along with the learner.

Animated Agent: Provides the learner with her first task-specific background information about plant anatomy, flying in with his jetpack to point out major parts of the leaf.

Animated Agent: Watches grimly as a leaf bursts open in the intense heat, while he explains, "Well, thick-skinned leaves just won't be able to give off enough water vapor to cool the plant in this hot climate. In order for the plant to transpire freely, the leaves should be thin-skinned." He then sits up and returns the initiative to the learner.

Learner: Considers the agent's advice but again proposes thick, thick-skinned leaves.

Because the learner has now repeatedly experienced difficulties with the abstract, conceptual advice, the agent determines that more direct advice is warranted and intervenes.

Animated Agent: Thin, thin, thin! Choose thin-skinned leaves.

Learner: Follows this direct advice then clicks on the *Done* button.

> If the learner instead had made another inappropriate leaf design decision, the agent would have taken the problem-solving initiative. After first exclaiming empathetically, "I know, sometimes this plant construction stuff can be really frustrating. But, that's when I help! Why don't you let me get this choice so we can move on to the next task. We may see hazards like this later on, on some other planet," he would then have performed an appropriate problem-solving action himself, and the leaves he created would have then been displayed in the design chamber. A final check is made to determine whether all tasks have been accomplished correctly, since the learner always has the option of shifting her attention from an incomplete task to work on one of the others. If there had been unresolved suboptimal design decisions on other sub-tasks, Herman would have offered advice at the appropriate level as described above, just as he would have done had the sub-task not been interrupted. Because all is well, the behavior sequencing engine directs the agent to exhibit an episode-completing congratulatory behavior.

Animated Agent: Cartwheels across the screen, teleports to the edge of the cliffs for a spectacular bungee jump, and then returns to introduce the next environment.

This article presents the computational mechanisms required to achieve precisely this style of mixed-initiative problem-solving interaction between lifelike pedagogical agents and learners.

3. Context Modeling for Mixed-Initiative Interaction

To facilitate mixed-initiative problem-solving, lifelike pedagogical agents must have access to a well-represented model of the problem-solving context. Specifically, to make decisions about which behaviors a pedagogical agent should perform, when they should be exhibited, and how they should be sequenced, the agent behavior sequencing engine maintains a dynamically updated design-centered contextual representation of mixed-initiative design episodes (Figure 3). In the figure, each V_i is an environmental variable, e.g., the amount of sunlight; each C_i is a component type of the artifact being designed, e.g., the roots; and each $SubTask_i$ is that of determining particular features for a given component type, e.g., selecting large, thin, and thick-skinned leaves. The arrows within the learning environment module represent the links between each component type and its related sub-task in the learner's current design episode. As the learner makes her design decisions, the behavior sequencing engine monitors the activity, updates the task model, selects behaviors from the behavior space, and assembles them into a multimodal behavior stream (as described in a section below). The design-centered context representation consists of an *environmental context*, a *multimodal advisory history*, and a *task model*:

– *Environmental Context:* Critical features of the environment which have been presented to the learner:

  - *Current Environment*: Environmental factors (and their values) in the current design episode.
  - *Environmental Intent*: Set of component types from the artifact library which that environment is intended to exercise.
  - *Environmental Complexity*: Associated with every environment is an environmental complexity that indicates the expected difficulty that learners will experience with problems in that environment.
  - *Sub-Task Complexity*: Associated with each sub-task for every environment is a complexity rating.
- *Multimodal Advisory History:* Critical features of the advisory dialogue, where each entry consists of:
  - *Topic*: Indicates environmental factors, artifact components, and constraint packets.
  - *Frequency Annotations*: Indicate the number of times that the learner has been advised about the topic(s).
  - *Media Annotations*: Indicate the media that were employed to communicate the advice.
- *Task Model:* Critical features of the task performed by the learner:
  - *Artifact-based Task Model*: Represents selection of component instances for the current artifact under construction, as well as a *focused component*.
  - *Design Evaluation*: When the learner completes a design, the artifact is evaluated as successful or not successful in the current environment.
  - *Problem-Solving Idle Time*: Time elapsed since the learner's last action.

3.1. Environmental Context

Design-centered problem solving revolves around a carefully orchestrated series of design episodes. To illustrate, consider design episodes in the domain of botanical anatomy and physiology. Learners are given an environment that specifies biologically critical factors in terms of qualitative variables. Environmental specifications, for example, might include the average incidence of sunlight, the amount of nutrients in the soil, and the height of the water table.

Learners consider these environmental conditions as they inspect components from a library of plant structures that is segmented into roots, stems, and leaves. Each component is defined by its structural characteristics such as length and amount of branching. Employing these components as building blocks, learners work in a "design studio" to graphically construct a customized plant that will flourish in the environment. Each iteration of the design process consists of the learners inspecting the library, selecting plant components to design a complete plant, and determining how the plant would then fare in the given environment. If they find that the plant would not survive, learners modify their plant's components to improve its suitability and the process continues until they have developed a robust plant that prospers in the environment.

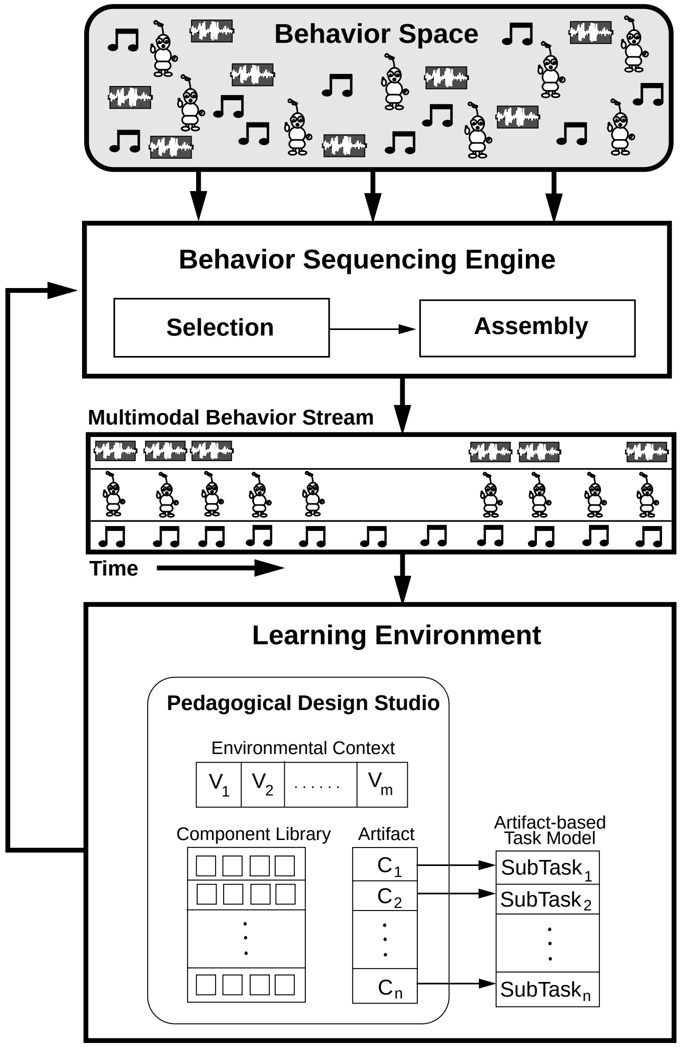

Figure 3. The Behavior Sequencing Engine, Design Studio and Artifact-Based Task Model

The environmental context guides the behavior sequencing engine's problem-solving advice by providing it with knowledge about the current environment and the pedagogical import of this environment. The *current environment* is encoded as a feature vector of active environmental features and their qualitative values. To illustrate, Table I depicts four representative environments from DESIGN-A-PLANT. The *environmental intent* represents the types of components the current environment is intended to exercise. For example, several of the environments in DESIGN-A-PLANT are intended to exercise learners' knowledge of leaves and the relation of leaf attributes to the features of the current environment, e.g., the

Table I. Sample environments from DESIGN-A-PLANT

Environment	*Feature*	*Value*
Desert Canyon	sunlight	low
	rain	low
	wind	high
	water table	low
Alpine Meadow	water table	high
	temperature	low
	rain	low
	wind	high
Tropical Cliffs	rain	low
	temperature	high
	nutrients	low
	water table	low
Southern Marsh	rain	high
	sunlight	low
	water table	high
	temperature	high

amount of available light. The *environmental complexity* represents an estimate of the expected relative difficulty that curriculum designers expect learners will experience in solving problems of that environment. Finally, each environment also includes *sub-task complexity* ratings which indicate expected difficulty levels for each sub-task; computationally, this is the number of component features for that sub-task which must be correctly selected to design a component for the environment. For example, in low rainfall environments, designing successful roots requires the learner to grapple with both `depth` and `branchiness` issues. All aspects of the environmental context are used in determining topics of advisory interventions.

To relate features of the environmental context to problem-solving actions, a constraint-based domain model can furnish the essential representational structures to support mixed-initiative pedagogical dialogue. For example, DESIGN-A-PLANT's domain model was developed for middle school students learning about botanical anatomy and physiology. It consists of constraint packets that relate environmental factors to plant components and the roles they play in plant physiology. In particular, these constraint packets encode the relationships between binary-valued environmental factors (e.g., incidence of sunlight, temperature, amount of nutrients in the soil), binary-valued features of the plant structures (roots, stems, and leaves), and the primary physiological processes (respiration, photosynthesis, osmosis, nutrient transport, water transport, and transpiration).

3.2. Multimodal Advisory History

To enable agents to provide learners with advice that is timely, that is delivered at appropriate levels of abstraction, and that employs appropriate media, the behavior sequencing engine requires a multimodal advisory history. Accordingly, the advisory history consists of a temporally ordered set of advice entries, which is updated the moment the agent provides advice. Each entry in the advisory context encodes three aspects of the advice. First, the *topic* of the advice represents the environmental factors, artifact components, and constraint packets about which the agent communicates. Second, agents need to take into account difficulties that the learner may be experiencing with a particular topic. Hence, *frequency annotations* indicate the number of times that the agent has advised the learner about the given topic(s) of the entry. Annotations on particular topics are suggestive of marks on an *overlay user model* (Carr and Goldstein, 1977); overlay marks indicate which subskills a learner has mastered, while frequency annotations indicate the topics about which the agent has advised the student.* Frequency annotations are used by the behavior sequencing engine to assess the level of abstraction at which advice should be provided, as discussed below. Third, because some advice is communicated with large-scale animated behaviors while others (e.g., reminders) are communicated primarily with verbal behaviors, the agent needs to be able to reason about the media that were employed to communicate the particular advice. *Media annotations* on each entry enable the behavior sequencing engine to reason about appropriate modes of expression. Together, the components of the multimodal advisory history are used to determine which topics of explanation and advice have been covered previously by the agent.

3.3. Task Model

Finally, of the three components of the design-centered context representation, the task model is the most critical to agents' pedagogical effectiveness. To make appropriate decisions about initiative control and behavior sequencing, the behavior sequencing engine requires an up-to-date representation of the task performed by the learner. The task model provides this knowledge by tracking the learner's problem-solving activities with an artifact-based task model, continuous evaluations of the viability of design decisions, and a problem-solving clock.

Dynamically providing goal-specific interventions requires the system to recognize learners' intent, but plan recognition is a notoriously difficult problem (Car-

* Higher frequency annotations are produced for topics with which the learner has experienced the most difficulty.

Table II. Instance of the Artifact-Based Task Model for DESIGN-A-PLANT

Subtask	*Design History*	*Current Subtask?*
Leaf Subtask		
Stem Subtask	Large, Thick, Woody	√
	Small, Thick	
	Small, Thin	
	Small, Thin, Green	
Root Subtask	Deep, Thick	
	Deep, Thin	
	Shallow, Thick	

berry, 1989; Chu-Carroll and Carberry, 1994; Hill and Johnson, 1995). To address the problem, artifact-based task models exploit (1) a well designed interface, e.g., learners interacting with the DESIGN-A-PLANT design interface (described below) signal their intentions through the normal course of problem solving, and (2) the "nearly decomposable" property of problems (Simon, 1981) to segment design tasks into sub-tasks $SubTask_1 \ldots SubTask_n$, where each $SubTask_i$ represents the sub-task of making a decision about components of a particular type, e.g., choosing between different types of leaves. Artifact-based task models encode three features:

– Each $SubTask_i$ records a history of design decisions made by the learner for that aspect of the design.
– Each completed $SubTask_i$ records the most recent design decision with the selected component (from C_i).
– Some sub-task $SubTask_f$ on which the learner is currently focused (the *focused component*) is marked.

To illustrate, suppose a learner interacting with DESIGN-A-PLANT has begun to solve the problems of which types of roots and stems to incorporate in her design for a particular environment. Furthermore, suppose she is currently considering issues bearing on stems, but has not yet begun to make decisions about leaves. The task model will be configured as shown in Table II, with the design history for each sub-task recorded in temporal order (most recent first).[⋆] Here, the learner has completed the `root` sub-task and the `stem` sub-task is currently in focus. The task model indicates that in considering this sub-task, her most recent decision was large, thick, woody stems.

[⋆] The top-to-bottom order of Leaf, Stem, and Root in Table II mirrors the spatial relation of tasks in the interface; in fact, the actual order in which the sub-tasks are performed is under the learner's control.

The artifact-based task model serves the critical function of providing a dynamically maintained design history, which informs the behavior sequencing engine's initiative control decisions. However, these decisions must take into account the quality of the learner's proposed designs and some measure of the learner's rate of progress and her engagement with the process. This supporting information is provided by (1) the *design evaluation*, which is supplied by a simple constraint system underlying the learning environment that determines whether a proposed design is appropriate for the given environmental conditions, and (2) the *problem-solving idle time*, which is computed by a running clock that tracks the amount of time that has elapsed since the learner's last design decision.

4. Dynamic Agent Behavior Sequencing for Mixed-Initiativity

Given a rich behavior space, the behavior sequencing engine (Figure 3) exploits the representations of the environmental context, the multimodal advisory history, and the task model to dynamically navigate through the behavior space, select agent behaviors, and assemble them in realtime, thereby enabling agents to engage in pedagogically effective mixed-initiative problem-solving episodes. Behavior sequencing for pedagogical agents is analogous to the topic sequencing that must be performed by pedagogical planners (Woolf and McDonald, 1984; Peachey and McCalla, 1986; Murray, 1990; Brusilovsky, 1992). Just as, for example, a discourse management network makes decisions about when to introduce a new problem, which topic to pursue next, and when to intervene, the behavior sequencing engine must also track learners' progress. Hence, behavior sequencing engines can exploit solutions to classic problems studied in the ITS community such as curriculum sequencing (Wescourt et al., 1981), simulation-based learning (Hollan et al., 1987; White and Frederiksen, 1987) and student modeling (Brown and Burton, 1978; Burton, 1982; Carr and Goldstein, 1977). However, behavior sequencing engines must also address new problems in orchestrating the agents' visual behaviors, coordinating visual behaviors with verbal behaviors, keeping the agent "alive" onscreen, and determining when visual, verbal, or both types of modes of intervention are appropriate. Below we describe the initiative control and intervention methods that address both the verbal and the visual modes of mixed-initiativity.

As learners solve problems, the behavior sequencing engine inspects the environmental context to identify the design criteria that the learner is attempting to satisfy and to determine which aspects of the design are most critical from a pedagogical perspective. It inspects the advisory history to track previously presented advice, and it inspects the task model to monitor the learner's progress and to note possible impasses. By extracting key features from these context models, using them together with ontological, intentional, and rhetorical indices to index into the behavior space, and dynamically sequencing the resulting behaviors, the behavior sequencing engine weaves together "local" behaviors from the behavior space to create "global" behaviors. These resulting behaviors enable the agent to share the initiative with the learner, to achieve pedagogical and visual coherence, and to appear lifelike.

At all times, the behavior sequencing engine maintains the agent's visual presence by keeping him* onscreen, visually immersed in the learning environment, and on or near the artifact which the learner is designing. In response to the learner's problem-solving activities, it directs the agent to provide problem-solving advice and to communicate fundamental knowledge of the domain to learners as they interactively design artifacts. After describing the types of behaviors critical for mixed-initiative interaction, the strategies and algorithms for controlling initiative, handling interventions, and selecting and assembling advisory, explanatory, and transition behaviors are presented below.**

4.1. Pedagogical Agent Behavior Categories

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

Pedagogical agents must be able to engage in a variety of explanatory, advisory, and believability-enhancing behaviors. Constructing a networked behavior space for an animated pedagogical agent capable of facilitating mixed-initiative problem solving entails specifying the following categories of behaviors:

- **Conceptual Explanatory Animated Behaviors:** The agent explicates the structures and functions of the artifact which is the subject of learning episodes. For example, the Design-A-Plant agent's behavior space contains an animated behavior of the agent explaining how root hairs absorb water through osmosis.
- **Problem-Solving Advisory Animated Behaviors:** The agent provides abstract, principle-based advice. Students must then operationalize this advice in their problem solving activities. For example, one animated behavior of the Design-A-Plant agent depicts him pointing out the relation between leaf size and low sunlight. (Plants in limited sunlight often have larger leaves.)

* Because the agent inhabiting the Design-A-Plant learning environment appears more masculine than feminine, we employ the masculine pronoun. Agents of course may be masculine, feminine, or indeterminate.

** Each of the algorithms described below has been employed in one or more versions of the implemented agent.

– **Problem-Solving Advisory Verbal Behaviors:** The agent provides abstract, principle-based advice as above, but in a verbal form.
– **Animated Transition Behaviors:** These portray the agent moving from one keyframe* to another keyframe, or performing an action that will set the stage for several upcoming behaviors.
– **Audio-Primary Problem Overviews:** The agent introduces a learner 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 learner 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."
– **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 this type of suggestion when a learner is experiencing serious difficulties.
– **Audio-Primary Interjections:** The agent remarks about the learner'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 learner 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 his stating, "It seems we're having some difficulty. Let's see if this helps . . . "
– **Believability-Enhancing Behaviors:** To enhance believability, the agent should perform a variety of physical actions. For example, the DESIGN-A-PLANT agent's Believability-Enhancing Behaviors include full motions such as reorientation (e.g., standing up, lying down) and smaller motions such as microbody movements (e.g., toe tapping, slight body shifts) and prop-based movements (e.g., glasses cleaning).

Informal empirical evidence from interactions of students with a variety of versions of the implemented agent suggests that each of the above behavior categories is necessary. For example, in the absence of Audio-Primary Advisory Reminders, the agent is forced to repeat (perhaps lengthy) advice again and again, rather than being

* A keyframe is a frame of an animation that represents a "still" of a character that serves as a reference position.

in a position to issue a less verbose reminder. While no claims are made about the sufficiency of these categories, it appears that each is necessary.

4.2. Mixed-Initiative Pedagogical Agent Behavior Sequencing

4.2.1. *Initiative Control*

To foster effective learning, we believe a key desirable feature of interaction of learners with pedagogical agents is that learners should be able to take control while they are performing problem-solving actions, and the agent should be able to take control when it appears that learners are experiencing difficulty or when they ask a question. After providing assistance, the agent should then relinquish control to learners so they may continue their problem solving.

To enact initiative transfers, the behavior sequencing engine operates in the following manner to transfer control back and forth from the agent A to the student S:

1. **Session Introduction:** At the beginning of a problem-solving session, the behavior sequencing engine directs A to introduce the learning environment. A *problem-solving session* consists of a series of *problem-solving episodes*. For example, in DESIGN-A-PLANT, learners travel from planet to planet, each with different environmental conditions for which they will design a plant.
2. **Episode Introduction:** At the beginning of each problem-solving episode, the behavior sequencing engine directs A to introduce the current problem.
3. **Relinquishing Control for Problem Solving:** Control is then transferred from A to S who undertakes problem-solving actions. Although S retains the initiative here, A may, nonetheless, perform Believability-Enhancing behaviors unless S is grappling with problems with high complexity.
4. **Yielding and Taking Advisory Initiative:** At any time, S may either request assistance or perform a problem-solving action. If (a) S requests assistance or (b) the intervention monitor determines that an intervention is warranted (see below), initiative is transferred to A, which becomes more alert visually, e.g., by sitting up or standing up, and then provides assistance according to the pedagogical behavior sequencing algorithms (described in detail below).
5. **Yielding and Taking Problem-Solving Initiative:** If after repeated attempts S demonstrates that she cannot solve the problem, A takes the initiative and performs the problem-solving action itself.
6. **Learner Problem-Solving Control Transfer:** After A has delivered relevant advice and explanations (or performed the problem-solving action), initiative control is returned immediately to S for continued problem solving.
7. **Episode Completion:** When S completes a problem-solving episode, the behavior sequencing engine directs A to exhibit a high-visual-impact congratulatory behavior. For example, in the DESIGN-A-PLANT environment, the agent cartwheels across the screen when learners successfully design a plant for an environment.

The pedagogical motivation underlying this initiative control scheme is straightforward. We wish to empower the learner to the greatest extent possible while at the same time providing a dialogue/problem-solving structure in which she may be the most successful. Hence, in contrast to a more didactic approach, we cede control to the learner immediately after the initial problem-solving introductions have been made and we intervene (see below) only when the problem-solving has reached an impasse.

4.2.2. *Task-Oriented Intervention*

A critical feature of controlling mixed-initiative interactions—particularly those that are to support learning—is *intervention*, where one of the interlocutors proactively takes control from another. Intervention decisions consist of determining the conditions under which an intervention should be performed and, if needed, determining the content of the intervention. The behavior sequencing engine monitors the state of the task model to assess when the learner requires assistance. If the learner makes an incorrect design decision (as indicated by her partial solutions), or if the problem-solving idle time exceeds a threshold, then the agent is directed to intervene. Empirical evidence with middle school students interacting with the DESIGN-A-PLANT learning environment indicates that the maximum period of "impasse" time without intervention for this age group should be approximately forty-five seconds.

In the general case for mixed-initiative design-centered problem-solving, the behavior sequencing engine must determine which component in artifact design its advice should address. This component selection strategy is especially significant for the "least invasive" learning environments. For example, in versions of the DESIGN-A-PLANT system in which the learner is free to make a number of design decisions before committing, the agent must determine which design decision to focus on. Hence, when the behavior sequencing engine determines that the agent should take the initiative, it determines the component C about which an interjection of advice (and possibly of explanations) should be provided according to the following prioritized strategy:

1. If the artifact-based task model's problem-solving idle time has exceeded its threshold, indicating the learner may be experiencing difficulty with the focused component C_f, the behavior sequencing engine directs the agent to provide advice about C_f.
2. If the task model indicates that the learner has just made an incorrect decision about a single component C, the behavior sequencing engine directs the agent to provide advice about C.
3. If the task model indicates that the learner has made incorrect decisions about multiple components $C_1 \ldots C_n$ of the artifact, the behavior sequencing engine inspects the focused component C_f (the component to which the learner is currently attending). If C_f is incorrect, the behavior sequencing engine directs

the agent to provide advice about it. (This clause comes into play in versions of DESIGN-A-PLANT when the learner is free to begin addressing other components before the component to which she was previously attending has been completed correctly.)

4. Otherwise, the behavior sequencing engine inspects the environmental intent* of the current environment and determines if one of the inappropriate components $C_1 \ldots C_n$ is the subject of the environmental intent. If so, it will provide advice about that C_i.
5. If no component satisfies any of these criteria, then the behavior sequencing engine randomly selects one of the inappropriate components C_i.

The design of the initiative control scheme and the prioritized intervention strategy were motivated by the desire to ensure that problem-solving episodes remain as coherent and "on-task" as possible. Hence, although there are numerous situations in which a more aggressive intervention strategy could seize the initiative, the alternate approach taken here seeks to strike a balance between the benefits of clearly (though perhaps verbosely) explaining fundamental domain knowledge on the one hand, and providing the learner the opportunity to absorb this knowledge in appropriate problem-solving contexts on the other hand. Therefore, with the exception of introductions to problems, the initiative control scheme strongly favors initiative being held for the most part by the learner.

4.2.3. *Advice Interjection*

Once the decision to intervene has been made and the component C on which the intervention will focus has been determined, the behavior sequencing engine must then determine how to provide the learner with appropriate advice: it must carefully orchestrate the agent's behaviors to ensure that the advice is properly constructed for the current problem-solving context. To do so, it inspects the current environment and the artifact-based task model to determine the active environmental features and the relevant artifact components. It then uses an *intentional index structure* to identify advisory topics that are germane to the current problem-solving situation. To do this, it employs the intentional indices to map the following problem-solving context variables onto relevant advisory topics $T_1 \ldots T_n$:

- C: The component selected above, e.g., `leaves`.
- $EnvtComplexity$: Associated with every environment is an environmental complexity that indicates the expected difficulty that learners will experience with problems in that environment. The *EnvtComplexity* for a particular environment is determined by curriculum designers by determining the number of active constraints in that environment. For example, one `level 2` environment requires plant designs to work under the conditions imposed by both a high watertable and a low ambient temperature.

* Recall that the environmental intent is the set of component types from the artifact library which that environment is intended to exercise, such as stems.

– $EnvtType$: Also associated with every environment is an $EnvtType$ indicating the type of the environment, e.g., `alpine`.

Each of these problem-solving context variables is critical for identifying appropriate topics of advice. C and $EnvtType$ guide the selection of advice that is relevant to the current design component. Using $EnvtComplexity$ assists the behavior sequencing engine in finding advice whose degree of sophistication (or simplicity) is appropriate. Indexing on these yields the relevant advisory topics $T_1 \ldots T_n$.

Finally, the value of $DesignEval$, which is the learning environment's evaluation of the particular componential solution proposed in the current sub-task, is used to select all of the relevant T_i that are helpful. $DesignEval$ indicates not merely a binary correctness vs. incorrectness evaluation, but also the correctness of individual features of the proposed solution. For example, leaves in the `alpine` environment in DESIGN-A-PLANT must have three correct features: `(size small)`, `(thickness thick)`, and `(skin-thickness thick)`. $DesignEval$ for a learner's proposed leaf structure in the alpine environment indicates correctness for each of these features. By employing this "feature-specific" variable $DesignEval$, we enable the agent to provide advice that is as specific as possible for the particular features of the learners' design decisions that may be problematic and, therefore, deserving of the agent's (and the learners') further attention.

4.2.4. *Selecting Stratified Problem-Solving Advice Levels*

A particularly critical decision to be made is to determine the *level* at which the advice should be delivered. Adopting the *knowledge compilation* view of learning, e.g., (Anderson, 1983; Newell, 1990), the pedagogical behavior sequencing algorithm is designed to provide learners with advice at a level appropriate for their mastery of the domain. An agent's high-level (indirect) advice provides assistance couched in terms of the knowledge contained in constraint packets, i.e., the functional relation between environmental factors and artifact components. An agent's low-level (direct) advice provides assistance couched in terms of very specific design decisions. While direct advice is easily operationalized, the opportunity for learning is reduced, so the algorithm gives preference to indirect advice.

This adaptive advisory strategy is captured in a stratification of advisory behaviors into four levels that represent varying degrees of directness. After the behavior sequencing engine has determined that the agent should provide advice about a particular topic T, it uses the following stratification of advisory behaviors relevant to T to interject advice that is cast at the appropriate level L. When it has been determined that advice about T should be delivered, it consults the multimodal advisory history to determine the previous L at which advice on T was delivered and then selects advisory behavior(s) A at $L - 1$:*

* Note that at runtime, the behavior sequencing engine first attempts the most abstract advice (`level 4`) and, if the learner continues to experience difficulty, gradually proceeds downwards through these stratification levels toward more direct advice and, eventually, action (`level 1`).

1. **Direct Action**
   - *Role:* After commenting about the difficulty of the problem, the agent performs the optimal problem-solving action himself.
   - *Features:* Selected as a last resort only after all problem-solving advice has failed.
   - *Example:* Herman intervenes by first explaining, “Wait, I know this one! Let me make this choice so we can get on to the next task. And, you may get a chance at this hazard again in some other environment.” He then performs the problem-solving action himself.
2. **Direct Verbal**
   - *Role:* The agent provides verbal advice that is direct and immediately operationalizable.
   - *Features:* Terse and does not require deduction; only provided after both forms of abstract advice have failed.
   - *Example:* In a low sunlight environment, Herman might say, “Make those leaves large.”
3. **Abstract Animated**
   - *Role:* The agent provides advice that is abstract and requires operationalization by the learner.
   - *Features:* Animated but indirect; provided only after abstract verbal advice proved ineffective. More visually distracting but clearer.
   - *Example:* In a low sunlight environment, Herman can appear suspended by his jetpack next to a small leaf. He can explain, “A plant with small leaves in dim sunlight cannot conduct enough photosynthesis and will have, ugh, no food!” With a wide smile, he demonstrates stretching the leaf out to a much larger surface area and tells the learner, “We can help this plant by giving it larger leaves; then it can do more photosynthesis and have plenty of food to eat.”
4. **Abstract Verbal**
   - *Role:* The agent provides the most abstract advice possible so the learner is required (if possible) to operationalize it.
   - *Features:* Terse and verbal, thereby requiring the greatest deduction and having a minimal visual distraction.
   - *Example:* In environments with low sunlight, which require learners to provide for increased photosynthesis, Herman might say, “In this environment, there’s not much sunlight. Remember that photosynthesis, the plant’s way of making food, occurs mostly in the leaves. Think about what types of leaves are gonna allow the plant to still make plenty of food even though there’s not much sunlight; large leaves or small leaves?”

4.2.5. *Explanation Interjection*

In addition to providing problem-solving advice, pedagogical agents can also facilitate learning by providing explanations of fundamental knowledge about the domain. By opportunistically interjecting explanations of domain phenomena that are relevant to (but not absolutely critical for) problem solving activities, pedagogical agents can broaden learners' knowledge in a situated manner. However, it is critical to interject this knowledge in a way that (a) is topical, (b) temporally distributes fundamental knowledge explanations in an even manner across the entire problem-solving session, and (c) obeys prerequisite requirements. If the behavior sequencing engine opted for abstract animated advice above, unless the current intervention was triggered by a problem-solving idle time violation—if a problem-solving idle time violation occurred, it is inferred that the learner is experiencing great difficulty and, therefore, that her attention should not be diverted by including auxiliary explanations—the behavior sequencing engine determines relevant animated explanatory behaviors E^P by performing the following computations:

1. **Select explanatory behaviors E that are relevant to the current problem-solving context.** Using an *ontological index structure*, it maps the selected component C to candidate explanatory behaviors E (Conceptual Explanatory Animated Segments) that are currently relevant.
2. **Compute m, 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.* The floor is taken for non-integer results to be conservative—representing the number of Conceptual Explanatory Animated Segments that should be exhibited.
3. **Select the subset E^P of not more than m explanatory behaviors E that are pedagogically viable.** We say that an explanatory behavior is *pedagogically viable* if (a) it has not been exhibited previously in this problem-solving session and (b) all of its prerequisite behaviors have already been exhibited. Explanatory behaviors are organized in a prerequisite structure, where 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. In general, 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.
4. **Mark the explanatory behaviors E^P.** Record in the multimedia dialog history that the selected behaviors have been exhibited.

* For example, f in the behavior sequencing engine of the current version of DESIGN-A-PLANT considers the efficiency with which the learner has reached the current level of environmental complexity ($EnvtComplexity$) and, from the number of remaining levels, estimates the number of environments left to be visited.

Each of the steps in the explanation interjection algorithm plays an important role in ensuring that the most relevant fundamental knowledge is presented. Step (1) eliminates explanations of fundamental knowledge that are not germane to the current component. Employing m in step (2) has the effect of evenly distributing these explanations over the course of the entire learning session. Because many domains and tasks are highly complex and learning time is limited, step (3) allows agents to take into account temporal resources to provide the greatest coverage of the domain in the given time. Finally, step (4) ensures that the multimedia dialogue history remains up-to-date.

To illustrate, the DESIGN-A-PLANT agent's behavior space currently includes five explanatory behaviors which the agent can exhibit to progressively reveal the rationale of the constraint relationships that drive leaf design. These are organized in a prerequisite ordering from explanations of macro-level leaf anatomy to micro-level leaf anatomy. For example, in one of the macro-level leaf anatomical explanations, he describes the blade of the leaf, the lamina, the petiole, and the midrib. In other leaf anatomy explanations he dives into the leaf, using his laser and a magnifier to show finer details. At deeper levels he takes the learner on a tour of cell anatomy, and at the deepest level, he provides a molecular explanation of photosynthesis chemistry.

Assuming that C was selected to be `leaf`, the behavior sequencing engine would first determine the relevant E (the Conceptual Explanatory Animated Segments) by identifying the leaf explanatory behaviors. Next, it would compute m by considering pedagogical factors in the following way: if the learner first has problems with leaf design at the pedagogical behavior sequencer's second level of complexity, the calculation of how many explanatory behaviors for the agent to exhibit uses the total number of explanatory behaviors in E (five), the learner's progress (three environments visited to date) and the number of levels remaining (two) to decide to request the agent to exhibit just one of the five. It chooses the subset E^P by identifying a behavior in E whose prerequisite behaviors have already been exhibited, which in this case is the first, most macro-level, explanatory behavior about gross leaf anatomy.

It is important to note that explanatory behaviors are invoked only when animated advice is invoked. Because explanatory behaviors typically involve visually sophisticated animations, they can convey more complex knowledge than purely verbalized advice; they are more powerful, but they are also more visually distracting. Consequently, the behavior sequencing engine is designed to maximize usage of bandwidth while simultaneously minimizing visual interruptions.

4.2.6. *Verbal and Visual Transitions*

In the same manner that coherence plays a critical role in assisting readers' comprehension of text (Grimes, 1975), the behaviors of lifelike pedagogical agents should be molded by considerations of both verbal and visual coherence. To achieve this, the behavior sequencing engine introduces both audio and visual transitions. Verbal transitions T_A are provided in the form of meta-comments about the learning episode and the agent's intention to set the stage for upcoming advice and explanations. Since all verbal behaviors are organized by rhetorical indices—in the current implementation, these are fairly coarse-grained and include Audio-Primary Problem Overviews, Audio-Primary Direct Suggestions, Audio-Primary Interjections, and Audio-Primary Transitions—the behavior sequencing engine notes that a verbal transition is called for and selects a verbal transition. For example, if by using the mechanisms above it has been deemed that advice is called for, Herman might first say, "OK, we're having some trouble here, but we're here to learn. Maybe this will help . . . "

Visual transitions are equally critical, both to help focus the learner's attention on animated explanations and to contribute to believability. Depending on the agent's current physical state within the learning environment (described below), the behavior sequencing engine will optionally select a *prefixing* visual transition $T_{V_{pre}}$ to anticipate animated explanations and a *postfixing* visual transition $T_{V_{post}}$ to conclude an animated explanation and yield the initiative to the learner. To determine whether $T_{V_{pre}}$ and $T_{V_{post}}$ behaviors are warranted and, if so, which ones, the behavior sequencing engine maintains spatial knowledge about the agent and adheres to physical constraints on movement for visual continuity. Spatial knowledge about the agent's current state is represented with a locational pair (P, O), where P symbolically (rather than geometrically via a coordinate-based representation) represents the agent's screen position and O represents its orientation, e.g., `(mid-bar-left, reclining)`, and physical constraints stipulate continuity relationships between behaviors. For example, the DESIGN-A-PLANT agent employs an orientation constraint: if the agent is standing, he cannot perform the lying down behavior; rather, he must first sit down before lying down.

Hence, if the behavior sequencing engine has determined that animated explanatory behaviors will be exhibited, it will also include visual and verbal transition behaviors. While adhering to the physical constraints, the behavior sequencing engine prepares a learner for an animated explanation by selecting a $T_{V_{pre}}$ that re-orients the agent into a more relaxed state (i.e., `sitting` if it is currently `standing`, `reclining` if it is currently `sitting`). In a similar fashion, it selects a $T_{V_{post}}$ by reversing this state. The net visual effect of these actions is that the agent appears to be relaxing to watch the animated explanations *with* the learner.

4.2.7. *Behavior Assembly and Presentation*

As it assembles the final set of behaviors determined above, the behavior sequencing engine must create a "global" behavior from each of the "local" behaviors in a manner that produces visually and pedagogically coherent actions in the learning context. It must transition the agent from a state of observing the learner's problem-solving, to a state of taking the initiative, to a state of holding the initiative during the intervention by exhibiting relevant explanatory and advisory behaviors, to the final state of returning the initiative to the learner. To accomplish these transitions, it imposes the following temporal ordering on the selected behaviors:

1. Verbal (audio) transition behavior T_A.
2. Prefixing visual transition behaviors $T_{V_{pre}}$.
3. Pedagogically viable explanatory behaviors E^P relevant to the selected component C, where $|E^P| \leq m$ and the behaviors are ordered by prerequisite structure.
4. Advisory behavior A about topics $T_1 \ldots T_n$, each with appropriate levels L_i and, consequently, each with appropriate mode (visual and/or auditory), that is relevant to the selected component C.
5. Postfixing visual transition behavior $T_{V_{post}}$.

The inclusion and ordering of each type of behavior play a critical role in the overall intervention. Introducing T_A behaviors first paves the way for upcoming explanations and advice; without them, the agent's behavior appeared abrupt. Inserting $T_{V_{pre}}$ behavior plays a similar role, but for the visual mode. Including E^P *before* A is very important. Pilot studies with learners interacting with different versions of the behavior sequencing engine suggested revealed that some arrangements of behaviors are considerably more effective than others. For example, in an earlier version of the system, the agent exhibited the A behaviors before the E^P behaviors: this ordering turned out to be problematic since learners tended to forget the advice offered in A because, we hypothesize, there were intervening conceptual explanations. The sequencing engine's assembly mechanism was therefore modified to present advisory behaviors after the explanatory behaviors. Finally, $T_{V_{post}}$ behaviors play an important role in visually signalling that the initiative has again shifted back to the learner.

The behavior sequencing engine directs the agent to immediately exhibit the resulting behaviors in the learning environment, and the process is repeated—monitoring followed by sequencing and assembly—as the learner continues to interactively solve problems in the environment. In addition, the agent continues to be attentive to the learner's activities through its physical positioning. For example, when learners interacting with the DESIGN-A-PLANT environment move the component task bar on which the agent is standing to the top level, the behavior sequencing engine directs the agent to shift to an orientation in which he does not bump his head on the top of the interface. The net effect of the sequencing engine's activities is rich problem-solving sessions where learners perceive they are interacting with an intriguing lifelike character who is attentively observing their problem-solving activities and actively providing customized multimedia advice.

5. Mixed Initiative Interaction: A Detailed Example

To illustrate the operation of the behavior sequencing algorithms, consider the DESIGN-A-PLANT learning scenario introduced earlier. A learner has watched Herman's Session Introduction, in which he presented explanations of elementary plant anatomy, and she has visited two planets. The first had a simple, high rainfall, environment which required her to choose thin leaves, for flexibility. In the second environment, a planet with dim sunlight and a low watertable, she needed assistance twice. She has now been escorted to a planet with low rainfall and high temperature.

> $EnvtType$ is now `tropical cliffs`, and $EnvtComplexity$ is `level 2`, indicating that the learner must attend to two environmental constraints during her problem solving here, and, because this environment focuses equally on all three tasks, the *environmental intent* includes `roots`, `stems`, and `leaves`. Because we have reached the beginning of a new problem-solving episode, the behavior sequencing engine directs the agent to introduce the current problem. Herman therefore exhibits the Audio-Primary Problem Overview associated with the current environment:

Animated Agent: Whoa! I'm feeling hot, hot, hot! Too bad there's no raindrops to fall upon my head. Well, the roots better be well chosen to soak up all the water they can. The stem and leaves still need to store as much water as possible, but at these high temperatures, they also need to be able to use some of that water to stay cool, by transpiration.

> Initiative is then transferred from agent to the learner who begins her problem solving. In general, learners can begin with any sub-task they wish.

Learner: Opts to begin plant construction with roots.

> Since the roots task is fairly complicated, as indicated by the *sub-task complexity* of `roots` in this environment, Herman is directed to avoid performing any potentially distracting Believability-Enhancing behaviors; rather, he stands quietly, attentively looking at the transparent chamber in which the learner is designing her plant.

Learner: Spends a while considering the rollover descriptions of the environment elements and roots settings icons, but cannot make a decision.

> Throughout learning sessions, the behavior sequencing engine tracks the *problem-solving idle time*, which has just reached 45 seconds. Because this period of time exceeds the threshold, the behavior sequencing engine directs Herman to take the initiative. To perform a task-oriented intervention, the behavior sequencing engine determines that the focused component (`roots`) is the component C about which advice should be provided. The behavior sequencing engine now uses the value of C, together with the current values of $EnvtType$ and $EnvtComplexity$ to index into the behavior space to determine the relevant advisory topics $T_1 \ldots T_n$. In this case, it determines that advice should be provided about two topics: T_1 will be advice about the effects of low rainfall on branchiness; T_2 will be advice about the effects of low rainfall on root depth. Next, it determines the level L at which to provide the advice. Because no advice has been provided before, it

gives advice at `level 4`, which is *Abstract Verbal*. Finally it determines that no explanatory advice should be provided because no explanatory behaviors are exhibited in response to an "exceeded problem-solving idle time" intervention. Because the advisory mode is verbal, no verbal transitions or visual transitions are needed.

Animated Agent: Because of the light rain in this environment, one of the choices you have to make is between branching and non-branching roots. Which type would increase the roots' contact with the scarce moisture?

Animated Agent: After a slight pause, raises a similar question about deep and shallow roots.

After completing his exhibition of the advisory behaviors, the agent returns the initiative to the learner.

Learner: Speculating that branching roots are more efficient and, deciding that shallow roots will, in Herman's words, "gather up the moisture as it soaks into the ground," chooses roots that are branching and shallow.

After the learning environment determines that the learner has made a valid design decision for the roots sub-task, $DesignEval$ is updated, and the behavior sequencing engine directs the agent to exhibit a congratulatory Audio-Primary Interjection for successful completion of the current sub-task and then returns the initiative to the learner.

Animated Agent: Congratulates the learner.

Learner: Raises the task bar to choose a stem.

The environment remains the same as above, but now the *sub-task complexity* is updated to indicate the difficulty of the `stem` task.

Learner: Mindful of Herman's comment about storing water in the stem and leaves, chooses a thick stem that is also short and covered with bark.

$DesignEval$ is now updated to reflect the fact that the learner has made a design decision that is valid with respect to the stem's thickness feature but not with respect to the bark decision. Because one of the features of the component currently under consideration will result in an artifact that cannot satisfy the current environmental constraints, the behavior sequencing engine takes the initiative from the learner and gives it to the agent. To perform a task-oriented intervention, it determines that the component C about which advice should be provided is the focused component, `stems`.

The behavior sequencing engine now uses the value of C, together with the current values of $EnvtType$ and $EnvtComplexity$ to index into the behavior space to determine the relevant advisory topics $T_1 \ldots T_n$. In this case, it determines that advice should be provided about a single topic T_2, namely, the environmental factors governing the presence or absence of bark on stems. Next, it determines the level L at which to provide the advice. Because no advice has been provided before, it gives advice at `level 4`, which is *Abstract Verbal*. Finally it determines that no explanatory advice should be provided (explanatory behaviors are exhibited only when the advice is animated) and therefore that no transitions are required.

Animated Agent: Whoo-eee! If the plant's in a hot area, there's a danger it could overheat. Like me, I'm so hot! To avoid this, the plant transpires, losing water through the stomata. Which will allow the plant to transpire more freely, a stem with bark on it or a stem with no bark?

Because the learner has demonstrated that she has understood one of two critical features of the current sub-task, the *sub-task complexity* is reduced. Noting this development, the behavior sequencing engine leaves the initiative with the learner but permits the agent to be more active. Even so, he waits a respectable amount of time before performing a self-absorbed behavior, such as polishing his glasses or deploying a yo-yo, as if to say "Don't forget I'm here to help, but you can take more time if you need to ..." Later, he will exhibit a random fidget or a toothy grin, as if to encourage the student to make a choice.

Learner: Correctly concludes that a stem with no bark will "transpire more freely."

After the learning environment determines that the learner has made a valid design decision for the stem sub-task, $DesignEval$ is updated, and the behavior sequencing engine directs the agent to exhibit a congratulatory Audio-Primary Interjection for successful completion of the current sub-task and then returns the initiative to the learner.

Animated Agent: Expresses his congratulations.

Learner: Shifts the focus to the leaves task.

When the learner was working on the preceding two tasks, Herman remained standing on the task bar. However, because the task bar is now high on the screen as the learner shifts to the leaves sub-task, the behavior sequencing engine leaves the initiative with the learner but at the same time directs Herman to sit down as the taskbar rises so as not to bump his head on the top of the screen. The environment remains the same as above, but now the *sub-task complexity* is updated to indicate the difficulty of the `leaves` task.

Learner: Knows to choose thick leaves for their ability to store the scarce moisture, but has not yet generalized her understanding of maximizing cooling by transpiration to choosing "thin-skinned" leaves; selects thick, large, thick-skinned leaves.

$DesignEval$ is now updated to reflect the fact that the learner has made a design decision that is valid with respect to the leaves' overall thickness feature but not with respect to the leaves' skin thickness decision. Because one of the features of the component currently under consideration will result in an artifact that cannot satisfy the current environmental constraints, the behavior sequencing engine takes the initiative from the learner and gives it to the agent. To perform a task-oriented intervention, it determines that the component C about which advice should be provided is the focused component, `leaves`.

The behavior sequencing engine now uses the value of C, together with the current values of $EnvtType$ and $EnvtComplexity$ to index into the behavior space to determine the relevant advisory topics $T_1 \ldots T_n$. In this case, it determines that advice should be provided about a single topic T_3, namely, the environmental factors governing leaf skin thickness. Next, it determines the level L at which to provide the advice. Because no advice has been provided before, it gives advice at `level 4`, which is *Abstract Verbal*. Finally it determines that no explanatory behaviors should be provided (explanatory behaviors are exhibited only when the advice is animated) and therefore that no transitions are required.

Animated Agent: In a hot climate, a plant needs to release excess water as vapor to cool itself off. Huh-huh. Basically it's sweating, just as we do, to prevent itself from overheating. Choose between those leaves which have a thick skin and those with a thin skin; which is better to keep the plant from overheating?

The behavior sequencing engine returns the initiative to the learner.

Learner: Makes a second incorrect design decision of thick, small leaves with thick skin.

$DesignEval$ is now updated to reflect the fact that the learner has made a design decision that is valid with respect to the leaves' overall thickness feature but not with respect to the leaves' skin thickness decision. Because one of the features of the component currently under consideration will result in an artifact that cannot satisfy the current environmental constraints, the behavior sequencing engine takes the initiative from the learner a second time by performing a task-oriented intervention about C, which is still `leaves`. It uses this value, together with the $EnvtType$ and $EnvtComplexity$ to index into the behavior space as before to determine that the relevant advisory topic, which is again T_3, skin thickness. However, this time the advisory level L is decremented to `level 3`, which is *Abstract Animated*. As a result, it now has an opportunity to interject animated explanatory behaviors.

Before it is ready to do so, however, it first selects explanatory behaviors E that are relevant to the selected component C (`leaves`), namely, all of the explanations of leaf anatomy. Next, it computes the m, the number of explanatory behaviors to exhibit. Because the number of environments remaining to be visited is high, the behavior sequencing engine determines that significant opportunities remain for providing explanations of leaf anatomy, so it chooses m to be 1. Pedagogical viability of the candidate behaviors is assessed by examining which explanations have both (a) not yet been exhibited (as indicated in the multimodal dialog history) and (b) have all of their prerequisites met. The explanatory behavior that satisfies these requirements E^P is the one in which the agent provides an introduction to external leaf anatomy, and finally E^P is then marked in the multimodal dialog history as having been exhibited.

Next, because animated advisory behaviors will be exhibited, an Audio-Primary Transition T_A is selected by choosing an introductory meta-comment. Finally, a prefixing visual transition $T_{V_{pre}}$ is selected in which the agent will sit down to watch the animations and a postfixing visual transition $T_{V_{post}}$ is selected in which the agent returns to his previous orientation. These behaviors are ordered as follows: T_A, $T_{V_{pre}}$, E^P, T_3, and $T_{V_{post}}$.

Animated Agent: OK, OK, so we're having some difficulty. But, that's OK, we're here to learn. I tell you what, see if this helps.

Animated Agent: Lies down on the task bar to watch the animations along with the learner.

A somber variation on the musical theme of this environment is playing.*

* Learning sessions in DESIGN-A-PLANT are accompanied by a context-sensitive soundtrack. In several experimental versions of the learning environment, the soundtrack composer provides thematic consistency of voicing and melody within a problem-solving episode and thematic consistency across problem-solving episodes. It exploits the task model to adapt its tempo, mood, and number of instrumental voices to the learner's progress.

Animated Agent: Provides the learner with her first task-specific background information about plant anatomy, flying in with his jetpack to point out major parts of the leaf.

Animated Agent: Watches grimly as a leaf bursts open in the intense heat, while he explains, "Well, thick-skinned leaves just won't be able to give off enough water vapor to cool the plant in this hot climate. In order for the plant to transpire freely, the leaves should be thin-skinned." He then sits up and returns the initiative to the learner.

Learner: Considers the agent's advice but again proposes thick, thick-skinned leaves.

> $DesignEval$ is now updated to reflect the fact that the learner has yet again made a design decision that is valid with respect to the leaves' overall thickness feature but not with respect to the leaves' skin thickness decision. It indexes into the advisory behavior space as before. This time, however, having exhausted the high-level, more abstract hints, the agent is forced to give more direct advice. Computationally, this is accomplished by decrementing the advisory level L to `level 2`, which is *Direct Verbal*. Because this advice is verbal, no auxiliary explanations are provided to accompany it and no transitions are required.

Animated Agent: Thin, thin, thin! Choose thin-skinned leaves.

Learner: Follows this direct advice then clicks on the *Done* button.

> If the learner instead had made another inappropriate leaf design decision, the behavior sequencing engine would have given control to the agent, who would then take the problem-solving initiative. This would have been accomplished by decrementing L to `level 1`, which is *Direct Action*. The agent would have been directed to say, "I know, sometimes this plant construction stuff can be really frustrating. But, that's when I help! Why don't you let me get this choice so we can move on to the next task. We may see hazards like this later on, on some other planet." The agent would then make the leaf design decision himself, and appropriately chosen leaves would then be displayed in the design chamber. A final check is made to determine whether all tasks have been accomplished correctly, since the learner always has the option of shifting her attention from an incomplete task to work on one of the others. If there had been unresolved suboptimal design decisions on other sub-tasks, Herman would have offered advice at the appropriate level as described above, just as he would have done had the sub-task not been interrupted. Because all is well, the behavior sequencing engine directs the agent to exhibit an episode-completing congratulatory behavior.

Animated Agent: Cartwheels across the screen, teleports to the edge of the cliffs for a spectacular bungee jump, and then returns to introduce the next environment.

6. Discussion

Lifelike pedagogical agents hold much promise for constructivist learning environments. Because of agents' potential pedagogical benefits and their lifelike qualities, they can play a critical role in mixed-initiative problem solving. However, assessing agent design decisions in the absence of a large body of empirical evidence is exceptionally difficult. Although we can abstractly formulate hypotheses about how to design behavior spaces, how to create representational structures for constructivist problem-solving contexts, and how to develop computational mechanisms for behavior sequencing engines, such conjecturing is unlikely to yield informative theories.

While work on lifelike pedagogical agents has just begun and our understanding of their design is therefore limited, during the past three years a number of "lessons learned" have begun to emerge from experiences with the DESIGN-A-PLANT agent. Through a series of iterative refinements consisting of design, implementation, and empirical evaluation, the agent has evolved from a stationary creature capable of providing only rudimentary assistance to a much more intriguing, lifelike character that monitors learners' progress, gives them helpful feedback, and gracefully intervenes in appropriate problem-solving contexts.

The primary impetus for the agent's successful evolution has been the findings of focus group studies. Conducted with more than twenty middle school students from Martin Middle School in Raleigh, North Carolina and with the Raleigh Chapter of the Women in Science Mentoring Program, these informal studies consisted of learners interacting with the agent for forty-five minutes to one hour. As each learner traveled with Herman from planet to planet, he or she solved design problems in DESIGN-A-PLANT environments. Learners were confronted with problems of varying levels of difficulty; some were very simple, involving only a single constraint, while others were complex, requiring learners to address multiple constraints simultaneously. As they designed plants for a variety of environmental conditions, the agent introduced problems, explained concepts in botanical anatomy and physiology, provided problem-solving advice, and interjected congratulatory and off-the-cuff remarks.

In general, Herman was unanimously well received by the students. His pedagogical and visual coherence, together with his immersive property—the fact that he inhabits the scenes of the environments to which learners travel—were perceived as strikingly lifelike behaviors. Herman's visual behaviors seemed to flow so well that no learner commented or displayed surprise during transitions. Because of the use of visual transition behaviors, initiative changes were for the most part visually flawless. His verbal reminders enabled learners to continue with their problem solving uninterrupted, and during the study learners made frequent (and unprompted) positive comments about his physical actions and remarks. The variety of his behaviors maintained their interest throughout the sessions, and most learners commented positively about the continuously updated score. Perhaps not surprisingly considering the middle-school audience, Herman's quirky asides were well received.

These studies suggest that lifelike pedagogical agents whose behaviors are selected and assembled with a well-designed sequencing engine can effectively guide learners through a complex subject in a manner that exhibits both pedagogical and visual coherence. The primary lessons gleaned from the studies are summarized below.

6.1. Rich Behavior Spaces for Mixed-Initiativity

To create mixed-initiative interactions, it is critical to populate an agent's behavior space with *at least* the nine types of behaviors identified earlier. Conceptual Explanatory Animated Behaviors and Problem-Solving Advisory Animated Behaviors constitute the core of an agent's repertoire. They provide the central means for communicating explanations of fundamental conceptual knowledge and for providing advice when interventions are required. Animated Transition Behaviors provide visual continuity, and an agent's verbal behaviors complement the visual behaviors. Audio-Primary Problem Overviews are important for introducing problems; without them, learners who have limited experience with a new learning environment may become confused. Audio-Primary Advisory Reminders and Audio-Primary Direct Suggestions provide multiple levels of advice. Audio-Primary Transitions provide rhetorical coherence. Agents' verbal meta-comments such as bridging phrases can also usher in topic transitions, and without them, agents' actions appear "choppy" and unmotivated. Audio-Primary Interjections and Believability-Enhancing Behaviors are essential for achieving the illusion of life.

6.2. Population-Specific Intervention Strategies

With appropriate intervention strategies, lifelike pedagogical agents can engage in effective mixed-initiative problem-solving interactions. By tracking changes in a task model representing the crucial problem-solving activities, the behavior sequencing engine can share the initiative with learners, enable them to interact freely with a learning environment to solve problems, take the initiative when assistance is warranted, and then relinquish the initiative as dictated by learners' progress. Intervention strategies should be motivated by an understanding of target user populations, problem-solving tasks, and domains. For example, observations of the students interacting with Herman during the focus group studies suggest that a relatively aggressive intervention strategy is perhaps most appropriate for design tasks for this user group. Although the agent told learners that they could ask for his help by clicking on him, in practice very few of the students in the focus group studies took advantage of the functionality, almost never requesting assistance. To address this, the intervention monitor was designed to be very sensitive to problem-solving difficulties. When learners make design decisions that violate environmental constraints, the agent immediately intervenes to assist them. In general, creating population-specific intervention strategies is critical to the whole mixed-initiative tutorial enterprise. Some target learner populations may need to be encouraged to explore the domain at their leisure, as in the classic work on microworlds where experimentation is a key component of the learning process (Cauzinille-Marmeche and Mathieu, 1988; Lawler and Lawler, 1987; Thompson, 1987). In contrast, in many training applications, concerns of efficiency prevail, so intervention must be conducted considerably more aggressively. Analogous differences obtain in different age groups as well. For example, adults can frequently help themselves after initially reaching an impasse, but children learning a new task sometimes get stuck in problem-solving "local minima" and require assistance in extricating themselves. We expect that it is for this reason that Herman's help was warmly received, but whether such an aggressive strategy is generally desirable is a subject for future studies.

6.3. Lightweight Task Models

To enable lifelike pedagogical agents to exhibit the flexibility required to assist learners in constructivist problem-solving, they must be provided with an up-to-date model of the problem-solving context. Perhaps no result from the last three years' experience is stronger than the following: without dynamically maintained task models that accurately reflect learners' problem-solving progress, lifelike pedagogical agents would be unable to engage in meaningful mixed-initiative problem-solving interactions. However, this does not imply that unusually expressive task models are required. In fact, the artifact-based task models employed in Design-A-Plant are of the "lightweight" variety. The task models are "lightweight" in somewhat the same sense that the user models in the Phelps just-in-time training system (Collins et al., 1997) are lightweight: while they are not particularly expressive, they are in practice highly accurate and can provide essential problem-solving tracking knowledge. Moreover, they permit non-invasive diagnosis. Because learners signal their intentions through interface actions that are observable by the task modeller, learners' problem solving can proceed uninterrupted. However, it is important to note that as the complexity of design tasks increase, the fairly simple non-invasive diagnostic techniques here may very well need to be replaced by those that are more invasive. For example, as we increase the degrees of freedom of the design tasks, the ability to create an interface that so clearly signals learners' intent may be reduced, and this may have the effects of, first, requiring a more "heavyweight" task model, and, second, forcing the agent to intervene more aggressively to ascertain learners' misconceptions.

6.4. Intervening with Multi-Level, Multimodal Advice

The problem-solving advice provided by lifelike pedagogical agents should have three interrelated properties: (1) it should be delivered at multiple levels of abstraction, (2) it should be delivered with media that are determined in a context-sensitive fashion, and (3) it should be carefully structured. The first property, multi-level advice, is a capability that is critical for agents which are intended to support knowledge compilation approaches to learning. High-level advice provides assistance that is couched in terms of more abstract domain concepts, e.g., the Design-A-Plant agent's high-level advice discusses the contents of constraint packets, while low-level advice is immediately operationalizable. By providing advice at these multiple levels, agents can attempt to foster knowledge compilation but will also have a fallback technique when learners experience difficulty with abstractions. The second property, context-sensitive media allocation, enables agents to effectively use both visual and auditory channels. They can perform more visually oriented behaviors for explaining new (and perhaps complex) concepts and more verbal behaviors for simple reminders. The former permits them to be significantly more expressive when needed, and the latter permits them to interject brief reminders and asides without distracting or disorienting learners. The third and final property, delivering advice that is carefully structured, was revealed by early focus group studies. In one of its early incarnations, the behavior sequencing engine first directed the agent to provide advice (advisory behaviors) and then to provide more conceptual explanations (explanatory behaviors) immediately prior to relinquishing the initiative so learners could return to their tasks. Because it was found that learners were confused by the agent first providing advice in response to a particular problem-solving impasse and then providing more conceptual knowledge, the initial behavior sequencing algorithms were modified to correct this problem by reversing the behavior orderings so that they conclude with advice.

6.5. Scaling Up Mixed-Initiative Tutorial Interactions

An important open issue in lifelike pedagogical agents is scalability. At this early stage in the research program, creating a lifelike pedagogical agent requires a fairly large investment in labor in terms of designing the entire approach to its behaviors, creating the behavior space, and building a behavior sequencing engine. While it is not conceivable that this effort could be reduced to zero in creating lifelike pedagogical agents for new tasks and domains, it has become clear that several techniques will contribute to scalability in the future. First, it has taken considerable effort to investigate, construct, and experiment with different intervention strategies, advice, and explanation. While this has been and will continue to be a critical research question, our understanding of how to create the overall design for lifelike agents has improved to a considerable degree from when the DESIGN-A-PLANT project began (hence, the current article). As we learn more about how mixed-initiative human-human tutorial interactions work in practice, the labor required for this aspect of the enterprise will be reduced substantially. Second, on a related topic, much of the early work on the DESIGN-A-PLANT agent was spent experimenting with different types of behavior spaces. While it seems that other types of behaviors will need to be identified to fill gaps not yet anticipated, much of this early exploratory work is now complete. Third, the behavior sequencing engine itself is a fairly complex system that currently requires an enormous amount of effort to iteratively design, construct, and refine. Although for "pedagogically sophisticated" lifelike agents, behavior control is likely to be an issue for some time to come, we expect that high-level tools for creating behavior sequencing engines will begin to appear in the not too distant future. In the same manner that a broad range of animation tools have brought multimedia to the general public, it seems likely that analogous authoring tools for lifelike pedagogical agents will enable instructional designers to create agents for mixed-initiative interaction on a cost-effective basis. Precisely what form these tools take and when they arrive remains to be seen.

7. Conclusions and Future Work

Lifelike pedagogical agents offer significant potential for mixed-initiative problem solving. Because they combine context-sensitive advisory behaviors with great visual appeal and they proactively assist learners performing exploratory problem-solving activities, they hold much promise for constructivist learning environments. In addition to their educational benefits, pedagogical agents with a strong lifelike presence may capture learners' imaginations and play a critical motivational role to keep them deeply engaged in problem solving.

We have proposed a computational framework for lifelike pedagogical agents that enables them to control the initiative in problem-solving interactions, achieve pedagogical and visual coherence, and exhibit believability. With a rich behavior space of animated and verbal behaviors, a behavior sequencing engine can exploit an environmental context, a multimodal advisory history, and an artifact-based task model to dynamically select and assemble an agent's behaviors in realtime. Focus group studies with middle school students interacting with an implemented agent in a fully functional constructivist learning environment suggest that lifelike pedagogical agents can contribute in important ways to constructivist learning. By taking advantage of a behavior space with ontological, intentional, and rhetorical indices and of dual pedagogical and believability-enhancing sequencers, a behavior sequencing engine can enable agents to provide context-specific multimedia advice while at the same time appearing lifelike and entertaining.

We believe that this work represents a promising first step toward creating lifelike pedagogical agents and that it consequently suggests a number of directions for future research. In particular, three lines of investigation are especially compelling: conducting formal empirical studies of pedagogical agents' effectiveness in learning environments; investigating the full spectrum of mixed-initiative interactions; and endowing pedagogical agents with full-scale realtime natural language generation and speech synthesis capabilities. These possibilities are discussed below.

First, as with all new learning environment technologies, full exploitation of lifelike pedagogical agents calls for a comprehensive research program to formally study their pedagogical and motivational effects. Results from initial studies of learner-agent interactions that were conducted with cognitive scientists have begun to emerge and are encouraging (Lester et al., 1997a; Lester et al., 1997b), but a significant body of work needs to be undertaken to determine precisely which intervention strategies, what types of behavior space representations, what task model representations, and which behavior sequencing algorithms are most effective in real-world classroom conditions. While the space of possible strategies, representations, and algorithms is enormous, we are optimistic that controlled empirical studies with learners interacting with multiple versions of the agent will demonstrate which design decisions are most effective in which situations. The DESIGN-A-PLANT agent and its learning environment will serve as a testbed for these types of studies.

Second, investigating the full spectrum of mixed-initiative interaction will reap important benefits for learner-agent problem solving. With significant advances in computational models of conversation-based, task-oriented dialogue (Walker, 1993; Smith and Hipp, 1994; Traum, 1994; Guinn, 1995; Freedman, 1996), we can expand the types of mixed-initiativity in which learners and agents can participate. For example, while the DESIGN-A-PLANT agent can provide a variety of types of explanations and advice, it cannot participate in complex dialogues requiring turn-taking, back channeling, or even rudimentary discourse segmentation. Extending its discourse functionalities to enable it to engage in "justification dialogues" in which learners could justify their design decisions would significantly improve its utility. As the discourse community continues to build a firm foundation for these capabilities and the quality of off-the-shelf speech recognition technologies increases, pedagogical agents can be extended to support considerably more sophisticated interactions with commensurate increases in pedagogical effectiveness.

Finally, providing pedagogical agents with full-scale realtime natural language generation and speech synthesis capabilities could significantly improve their flexibility in mixed-initiative interaction. For example, if the DESIGN-A-PLANT agent, which now employs vocal behaviors created by a voice actor, could employ the full arsenal of natural language generation techniques, it could exploit the generativity of natural language to provide advice whose content, discourse structure, phrase structure, lexicalizations, and prosody were carefully tailored to individual learners in much the same manner that human-tutorial dialogues are. Creating these capabilities will entail incorporating state-of-the-art explanation generation techniques (Suthers, 1991; Cawsey, 1992; Hovy, 1993; Mittal, 1993; Moore, 1995; Lester and Porter, 1997) and surface generators (Elhadad, 1991) and then extending them to take into account conversational, gestural, and deictic aspects of discourse (Cassell et al., 1994; Towns et al., 1998).

In summary, it appears that lifelike pedagogical agents have much to offer and that much remains to be done to bring them to fruition.

Acknowledgements

The authors wish to thank all of the members of the North Carolina State University IntelliMedia Initiative for their invaluable contributions to the DESIGN-A-PLANT project. Special thanks are extended to the following individuals and organizations: Patrick FitzGerald, Director of the School of Design's branch of the IntelliMedia Initiative; the implementation team, particularly Robert Stevenson and Michael

O'Leary, for their work on the behavior sequencing engine; the 3D modelers and animators, particularly John Myrick, James Dautremont, and Philip McKay, for their work on the animated agent, plant models, and environments; Chris Tomasson and her seventh grade class at Martin Middle School and the Raleigh Chapter of the Women in Science Mentoring Program for participating in the focus group studies; Stuart Towns and Charles Callaway for comments on earlier drafts of this article; the anonymous reviewers, whose insightful comments significantly improved the article. Support for this work was provided by the following organizations: 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; Novell, Inc.; and equipment donations from Apple and IBM.

References

Anderson, J., A. Corbett, K. Koedinger, and R. Pelletier: 1995, 'Cognitive Tutors: Lessons Learned'. *Journal of the Learning Sciences* **4**(2), 167–207.

Anderson, J. R.: 1983, *The Architecture of Cognition*. Harvard University Press.

André, E., W. Finkler, W. Graf, T. Rist, A. Schauder, and W. Wahlster: 1993, 'WIP: The Automatic Synthesis of Multi-Modal Presentations'. In: M. T. Maybury (ed.): *Intelligent Multimedia Interfaces*. AAAI Press, Chapt. 3.

André, E. and T. Rist: 1996, 'Coping with Temporal Constraints in Multimedia Presentation Planning'. In: *Proceedings of the Thirteenth National Conference on Artificial Intelligence*. pp. 142–147.

Bates, J.: 1994, 'The Role of Emotion in Believable Agents'. *Communications of the ACM* **37**(7), 122–125.

Blumberg, B. and T. Galyean: 1995, 'Multi-Level Direction of Autonomous Creatures for Real-Time Virtual Environments'. In: *Computer Graphics Proceedings*. pp. 47–54.

Brown, J., A. Collins, and P. Duguid: 1989, 'Situated Cognition and the Culture of Learning'. *Educational Researcher* **18**(1), 32–42.

Brown, J. S. and R. R. Burton: 1978, 'Diagnostic Models for Procedural Bugs in Basic Mathematical Skills'. *Cognitive Science* **2**, 155–191.

Brusilovsky, P.: 1992, 'A Framework for Intelligent Knowledge Sequencing and Task Sequencing'. In: *Proceedings of the Second International Conference, ITS '92*. pp. 499–506.

Burton, R. R.: 1982, 'Diagnosing Bugs in a Simple Procedural Skill'. In: D. Sleeman and J. S. Brown (eds.): *Intelligent Tutoring Systems*. London: Academic Press.

Carberry, S.: 1989, 'Plan Recognition and Its Use in Understanding Dialog'. In: A. Kobsa and W. Wahlster (eds.): *User Models in Dialog Systems*. Berlin: Springer-Verlag, pp. 133–162.

Carbonell, J. R.: 1970, 'AI in CAI: An Artificial-Intelligence Approach to Computer-Assisted Instruction'. *IEEE Transactions on Man-Machine Systems* **4**, 190–202.

Carr, B. and I. P. Goldstein: 1977, 'Overlays: A Theory of Modelling for Computer Aided Instruction'. Technical Report AI Memo 406, Massachusetts Institute of Technology, Artificial Intelligence Laboratory.

Cassell, J., C. Pelachaud, N. Badler, M. Steedman, B. Achorn, T. Becket, B. Douville, S. Prevost, and M. Stone: 1994, 'Modeling the Interaction between Speech and Gesture'. In: *SIGGRAPH '94*.

Cauzinille-Marmeche, E. and J. Mathieu: 1988, 'Experimental Data for the Design of a Microworld-Based System for Algebra'. In: H. Mandl and A. Lesgold (eds.): *Learning Issues for Intelligent Tutoring Systems*. New York: Springer-Verlag, pp. 278–286.

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

Chu-Carroll, J. and S. Carberry: 1994, 'A Plan-Based Model for Response Generation in Collaborative Task-Oriented Dialogues'. In: *AAAI-94: Proceedings of the Twelfth National Conference on Artificial Intelligence*, Vol. 1. pp. 799–805.

Cohen, R., C. Allaby, C. Cumbaa, M. Fitzgerald, K. Ho, B. Hui, C. Latulipe, F. Lu, N. Moussa, D. Pooley, A. Qian, and S. Siddiqi: 1998, 'What is Initiative'. *User Modeling and User-Adapted Interaction*. in this issue.

Collins, J., J. Greer, V. Kumar, G. McCalla, P. Meagher, and R. Tkatch: 1997, 'Inspectable User Models for Just-In-Time Workplace Training'. In: *Proceedings of the Sixth International Conference on User Modeling*. pp. 327–337.

Elhadad, M.: 1991, 'FUF: The Universal Unifier User Manual Version 5.0'. Technical Report CUCS-038-91, Department of Computer Science, Columbia University.

Feiner, S. K. and K. R. McKeown: 1990, 'Coordinating Text and Graphics in Explanation Generation'. In: *Proceedings of the Eighth National Conference on Artificial Intelligence*. Boston, MA, pp. 442–449.

Freedman, R. K.: 1996, 'Interaction of Discourse Planning, Instructional Planning and Dialogue Management in an Interactive Tutoring System'. Ph.D. thesis, Northwestern University.

Green, N. and S. Carberry: 1998, 'A Computational Mechanism for Initiative in Answer Generation'. *User Modeling and User-Adapted Interaction*. in this issue.

Grimes, J. E.: 1975, *The Thread of Discourse*. The Hague: Mouton.

Guinn, C. I.: 1995, 'Meta-Dialogue Behaviors: Improving the Efficiency of Human-Machine Dialogue – A Computational Model of Variable Initiative and Negotiation in Collaborative Problem-Solving'. Ph.D. thesis, Duke University.

Hale, C. R. and L. W. Barsalou: 1995, 'Explanation Content and Construction During System Learning and Troubleshooting'. *The Journal of the Learning Sciences* **4**(4), 385–436.

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

Hollan, J. D., E. L. Hutchins, and L. M. Weitzman: 1987, 'STEAMER: An Interactive, Inspectable, Simulation-Based Training System'. In: G. Kearsley (ed.): *Artificial Intelligence and Instruction: Applications and Methods*. Reading, MA: Addison-Wesley, pp. 113–134.

Hovy, E. H.: 1993, 'Automated Discourse Generation Using Discourse Structure Relations'. *Artificial Intelligence* **63**, 341–385.

Kurlander, D. and D. T. Ling: 1995, 'Planning-Based Control of Interface Animation'. In: *Proceedings of CHI '95*. pp. 472–479.

Lawler, R. W. and G. P. Lawler: 1987, 'Computer Microworlds and Reading: An Analysis for Their Systematic Application'. In: R. W. Lawler and M. Yazdani (eds.): *Artificial Intelligence and Education*, Vol. 1. Norwood, NJ: Ablex, pp. 95–115.

Lesgold, A., S. Lajoie, M. Bunzo, and G. Eggan: 1992, 'SHERLOCK: A Coached Practice Environment for an Electronics Trouble-Shooting Job'. In: J. H. Larkin and R. W. Chabay (eds.): *Computer-Assisted Instruction and Intelligent Tutoring Systems: Shared Goals and Complementary Approaches*. Hillsdale, NJ: Lawrence Erlbaum, pp. 201–238.

Lester, J. C., S. A. Converse, S. E. Kahler, S. T. Barlow, B. A. Stone, and R. Bhogal: 1997a, 'The Persona Effect: Affective Impact of Animated Pedagogical Agents'. In: *Proceedings of CHI'97 (Human Factors in Computing Systems)*. pp. 359–366.

Lester, J. C., S. A. Converse, B. A. Stone, S. E. Kahler, and S. T. Barlow: 1997b, 'Animated Pedagogical Agents and Problem-Solving Effectiveness: A Large-Scale Empirical Evaluation'. In: *Proceedings of Eighth World Conference on Artificial Intelligence in Education*. pp. 23–30.

Lester, J. C., P. J. FitzGerald, and B. A. Stone: 1997c, 'The Pedagogical Design Studio: Exploiting Artifact-Based Task Models for Constructivist Learning'. In: *Proceedings of the Third International Conference on Intelligent User Interfaces*. pp. 155–162.

Lester, J. C. and B. W. Porter: 1997, 'Developing and Empirically Evaluating Robust Explanation Generators: The KNIGHT Experiments'. *Computational Linguistics* **23**(1), 65–101.

Lester, J. C. and B. A. Stone: 1997, 'Increasing Believability in Animated Pedagogical Agents'. In: *Proceedings of the First International Conference on Autonomous Agents*. pp. 16–21.

Lester, J. C., B. A. Stone, M. A. O'Leary, and R. B. Stevenson: 1996, 'Focusing Problem Solving in Design-Centered Learning Environments'. In: *Proceedings of the Third International Conference on Intelligent Tutoring Systems*. pp. 475–483.

Maes, P., T. Darrell, B. Blumberg, and A. Pentland: 1995, 'The ALIVE System: Full-body Interaction with Autonomous Agents'. In: *Proceedings of the Computer Animation '95 Conference*.

Mark, M. A. and J. E. Greer: 1995, 'The VCR Tutor: Effective Instruction for Device Operation'. *Journal of the Learning Sciences* **4**(2), 209–246.
Maybury, M. T.: 1991, 'Planning Multimedia Explanations Using Communicative Acts'. In: *Proceedings of the Ninth National Conference on Artificial Intelligence*. Anaheim, CA, pp. 61–66.
Mittal, V., S. Roth, J. D. Moore, J. Mattis, and G. Carenini: 1995, 'Generating Explanatory Captions for Information Graphics'. In: *Proceedings of the International Joint Conference on Artificial Intelligence*. pp. 1276–1283.
Mittal, V. O.: 1993, 'Generating Natural Language Descriptions with Integrated Text and Examples'. Ph.D. thesis, University of Southern California.
Moore, J. D.: 1995, *Participating in Explanatory Dialogues*. MIT Press.
Murray, W. R.: 1990, 'A Blackboard-Based Dynamic Instructional Planner'. In: *Proceedings of the Eighth National Conference on Artificial Intelligence*. pp. 434–441.
Newell, A.: 1990, *Unified Theories of Cognition*. Harvard University Press.
Noake, R.: 1988, *Animation Techniques*. London: Chartwell.
Papert, S.: 1991, 'Situating Constructionism'. In: I. Harel and S. Papert (eds.): *Constructionism*. Norwood, NJ: Ablex, pp. 1–12.
Peachey, D. R. and G. I. McCalla: 1986, 'Using Planning Techniques in Intelligent Tutoring Systems'. *International Journal of Man-Machine Studies* **24**, 77–98.
Piaget, J.: 1954, *The Construction of Reality in the Child*. New York: Basic Books.
Rich, C. and C. L. Sidner: 1998, 'COLLAGEN: A Collaboration Manager for Software Interface Agents'. *User Modeling and User-Adapted Interaction*. in this issue.
Rickel, J. and L. Johnson: 1997, 'Intelligent Tutoring in Virtual Reality: A Preliminary Report'. In: *Proceedings of the Eighth World Conference on AI in Education*. pp. 294–301.
Roth, S. F., J. Mattis, and X. Mesnard: 1991, 'Graphics and Natural Language as Components of Automatic Explanation'. In: J. W. Sullivan and S. W. Tyler (eds.): *Intelligent User Interfaces*. New York: Addison-Wesley, pp. 207–239.
Simon, H. A.: 1981, *The Sciences of the Artificial*. Cambridge, MA: MIT Press.
Smith, R. W. and D. R. Hipp: 1994, *Spoken Natural Language Dialog Systems*. Cambridge, Massachusetts: Oxford University Press.
Stone, B. A. and J. C. Lester: 1996, 'Dynamically Sequencing an Animated Pedagogical Agent'. In: *Proceedings of the Thirteenth National Conference on Artificial Intelligence*. pp. 424–431.
Suthers, D. D.: 1991, 'A Task-Appropriate Hybrid Architecture for Explanation'. *Computational Intelligence* **7**(4), 315–333.
Thompson, P. W.: 1987, 'Mathematical Microworlds and Intelligent Computer-Assisted Instruction'. In: G. Kearsley (ed.): *Artificial Intelligence and Instruction: Applications and Methods*. Reading, MA: Addison-Wesley, pp. 83–109.
Towns, S., C. Callaway, J. Voerman, and J. Lester: 1998, 'Coherent Gestures, Locomotion, and Speech in Life-Like Pedagogical Agents'. In: *Proceedings of the Fourth International Conference on Intelligent User Interfaces*. pp. 13–20.
Traum, D.: 1994, 'A Computational Theory of Grounding in Natural Language Generation'. Ph.D. thesis, University of Rochester.
Walker, M.: 1993, 'Informational Redundancy and Resource Bounds in Dialogue'. Ph.D. thesis, University of Pennsylvania.
Wenger, E.: 1987, *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*. Los Altos, California: Morgan Kaufmann.
Wescourt, K. T., M. Beard, and A. Barr: 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*. Stanford, California: Stanford University, pp. 817–839.
White, B. Y. and J. R. Frederiksen: 1987, 'Qualitative Models and Intelligent Learning Environments'. In: R. W. Lawler and M. Yazdani (eds.): *Artificial Intelligence and Education*, Vol. 1. Norwood, New Jersey: Ablex, pp. 281–305.
Woolf, B. and D. D. McDonald: 1984, 'Building a Computer Tutor: Design Issues'. *IEEE Computer* **17**(9), 61–73.

Authors' Vitae

James C. Lester

Dr. James C. Lester is Assistant Professor of Computer Science at North Carolina State University, and founder and director of the IntelliMedia Initiative at NC State's College of Engineering. Dr. Lester received his B.A. degree in History from Baylor University and his B.A. (Phi Beta Kappa), M.S.C.S., and Ph.D. degrees in Computer Sciences from the University of Texas at Austin. Dr. Lester has worked in several areas of artificial intelligence including computational linguistics (discourse planning and realization in natural language generation), knowledge-based learning environments (animated pedagogical agents and 3D learning environments), and intelligent multimedia systems (intelligent virtual camera planning and coordinated 3D animation/speech generation).

Brian A. Stone

Brian A. Stone is a Ph.D. candidate in Computer Science at North Carolina State University. He received his B.S. degree in Mathematics and Computer Science from Carnegie Mellon University in 1992 and his M.S. degree in Computer Science from North Carolina State University in 1995. His primary interests in artificial intelligence lie in the areas of machine learning (neural networks) and knowledge-based learning environments (animated pedagogical agents).

Gary D. Stelling

Dr. Gary D. Stelling is an M.S. student in Computer Science at North Carolina State University. Dr. Stelling received his B.A. in Chemistry from Washington University in 1965, his Ph.D. degree in Organic Chemistry from Stanford University in 1970, and his B.S. degree in Data Processing from Washington University in 1988. His research in chemistry focused on emulsion polymerization, and his current research in artificial intelligence lies in knowledge-based learning environments (animated pedagogical agents).

Contents

List of Figures

List of Tables