Abstract

STELLING, GARY DEAN. Affective Behavior Control for Lifelike Pedagogical Agents. (Under the direction of James C. Lester.)

Lifelike pedagogical agents should be especially effective in *constructivist* learning environments in which students participate in active problem solving. We can simulate such a constructivist setting with personal computing using a well-designed, evocative graphical interface and the rich multimedia -- audio, video and animation -- currently available. Beyond such an authentic problem-solving context, constructivist learning employs a social aspect, centered on the interaction of learner and mentor. We submit that an animated pedagogical agent who delivers contextualized problem-solving advice can play the part of the expert. Further, we propose that an added measure of believability and motivation would result from giving these agents the ability to express situationally appropriate emotions. To test the promise of such an *affective* agent, we first identified the cognitive emotion types most useful in a problem-solving dialog. We then devised a structure to store the details of the learner's situation in order to determine the appropriate emotion from the pedagogical agent. These enhancements have been instantiated in a full-scale implementation of the lifelike pedagogical agent of DESIGN-A-PLANT, a learning environment developed in the domain of botanical anatomy and physiology for middle-school students. Evaluation by a focus group of students was encouraging. They preferred the emotional version of the agent and reported that his affective behavior was helpful in their problem solving.

AFFECTIVE BEHAVIOR CONTROL
FOR
LIFELIKE PEDAGOGICAL AGENTS

by
GARY DEAN STELLING

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

COMPUTER SCIENCE

Raleigh

2002

APPROVED BY:

Chair of Advisory Committee

Biography

Gary Stelling was born in 1943. On a small farm near Alva, Oklahoma, his family grew wheat; it was a marginal existence. Eventually, his father found more reliable work as a maintenance man and carpenter in St. Louis.

In high school, Gary pursued an early interest in the physical sciences, to which Sputnik injected a patriotic motivation. In college, he gradually narrowed this interest to organic chemistry, then as much art as technology. A helpful teacher got him into an excellent graduate school. There, apparently not the brightest and certainly not the most knowledgeable, he still got by with adventurous empiricism and uncommon creativity. His research on lifelike chemical reactions included a detour into the serious computing of X-ray crystallography.

Unsuited to teach and incapable of the encyclopedic knowledge needed for pharmaceuticals, he nevertheless was successful doing creative, resourceful polymer research for large companies, inventing processes for static-proofed carpet, disposable fabrics (such as diapers and surgical drapes) of low toxicity, specialized paper coatings and high-performance latex paints.

Meanwhile, Gary had been planning for early retirement as a hardware store owner. Instead, the hardware of microcomputing captured his imagination -- hardware that responded! In night school he managed a BS in Data Processing. Still later, he spurned industrial research for full-time study of Computer Science. Now he expects to complete his formal employment as a systems administrator, helping silicon-based and carbon-based intelligences network productively.

Acknowledgments

In the readings for their course on multimedia and knowledge-based learning environments (KBLEs), James Lester and Pat FitzGerald prominently included an early history of the Disney style of animation, The Illusion of Life. That title, I gradually realized while working with these teachers and their devoted students on several projects, was a fitting encapsulation of the goal of their IntelliMedia Initiative. It has proved to be a powerfully cohesive vision for all the participants; the AI / multimedia programmers along with the artists / designers were compelled to create the illusion of life, intelligence and responsiveness in educational software intertwined with imagery possessed of lifelike movement and expressiveness.

IntelliMedia colleagues / friends / mentors, including but never limited to Alex, Tim, Dennis, Heather, Charles, Joel, Mike, Stuart, Wes, William, Brent, Rob, Jennifer, Ann-Elise, Francis, Amanda, Will, Roberta, Luke, Michelle, Scott, Mary, I am humbly grateful to you for including me in your adventures.

This work is also indebted to the original and continuing members of the Design-A-Plant project, from its inception as the project in the first Multimedia and KBLE class through its extensive testing in several Middle School settings; especially, to Brian Stone, the brains and voice of Herman the Bug, and to John Myrick, Herman's heart and soul, for all their help as I tried to follow through with their vision of a meaningful and memorable character. Finally, especially, Mike Cuales drew out Herman's feelings.

Thanks to my committee members, for their example, patience and indulgence, and helpful comments.

To Pat, for his ability to highlight creativity, regardless of how well hidden, and to promote progress, no matter how gradual.

To James, for being an accepting, optimistic, tireless teacher and a far-sighted advisor.

To the students of Raleigh's Ligon GT Magnet Middle School who cheerfully checked out my emotional Herman, to their dedicated teachers Ann Thompson and Skip Thibault who coordinated the informal testing, to Principal Steve Takacs who always made room for NC State researchers.

And, especially, to my wife Nancy, for her encouragement, perspective and enduring sense of humor.

Support for this work came from NSF, the NC State IntelliMedia Initiative and Novell. Equipment donations by Apple and IBM.

Contents

List of Figures vii

1 Introduction 1

1.1 Pedagogical Agents 2

1.2 Thesis Overview 3

2 Learning Environment 7

2.1 Architecture 8

2.2 Design-A-Plant 17

2.2.1 The Domain 17

2.2.2 The Challenges 20

3 Behavior Control 22

3.1 Emotions 23

3.2 Programming Emotions 26

3.3 Fitting Emotions into DAP Architecture 31

4 An Implemented Agent 38

4.1 Example Design-A-Plant Session 40

4.2 Learning Environment Implementation 47

4.3 Pedagogical Agent Implementation 53

5 Evaluation 59

5.1 Context 59

5.2 Findings 61

6 Related Work 63

6.1 Design-A-Plant 63

6.2 Emotion Programming 70

6.3 Affective Pedagogical Agents 72

7 Conclusions and Future Work 76

7.1 Future Work 77

7.2 Concluding Remarks 80

Bibliography 82

List of Figures

Figure 1.1 Herman The Bug waits as a Design-A-Plant episode begins. 4

Figure 2.1 Learning Environment Architecture 9

Figure 3.1 Emotion types from Elliott's The Affective Reasoner 24

Figure 3.2 Inheritance of slots in a construal frame (The Affective Reasoner) 28

Figure 3.3 Emotion Eliciting Condition relation (The Affective Reasoner) 29

Figure 3.4 Affect Base hierarchy 30

Figure 4.1 Herman doesn't like this new planet. 40

Figure 4.2 Herman is worried. 41

Figure 4.3 Herman "was afraid of this." 43

Figure 4.4 Herman is hopeful. 44

Figure 4.5 Herman's optimism was rewarded. 45

Figure 4.6 Herman gives a Learner well-deserved, positive feedback. 47

Preface

The promise of educational software seems largely unfulfilled.

It is, after all, very hard work, to do right. Harder than textbook writing, the last successful new pedagogical technology. Yet we raise up the hope of well-designed learning environments that utilize the inherent flexibility of artificial intelligence and the richness of multimedia.

Hard work cubed.

But, hard work in the pursuit of lofty goals can be enjoyable! This thesis describes some of this inherently rewarding travail.

Chapter 1

Introduction

Lifelike pedagogical agents should be especially effective in constructivist learning environments [Lester *et al*. 1999]. Constructivism [Piaget 1954] has begun to dominate education (that is, instead of didactic pedagogy). Constructivist learning is the more customer-driven; its focus is the learner. The learner’s knowledge grows as they “construct their own understandings and capabilities in carrying out challenging tasks” [Collins 1991] (as referenced in [Gance 2002]). Piaget, on whose writings constructivism is based, is said [Doll 1993] to have felt that students have to be encouraged out of their inertia, that is, out of past patterns, with a “disequilibrium” stirred up in their environment by their teacher, without its becoming overwhelming.

The “virtual design studio” [FitzGerald & Lester 1997] that is the testbed of the research in this thesis, seems supremely constructivist and some work had already been done to make its pedagogical agent more “believable.” Pedagogical agents with a tendency to display contextually appropriate emotions would be more likely to "shake up" their students, to introduce Piaget's disequilibrium.

Before proceeding, it is important to reflect on what an agent is, fundamentally and in our context. The broader meaning would be "one empowered to act for or represent another" and our meaning would be close to "that which has the power to produce an effect; as, a physical, chemical, or medicinal agent."

And, "lifelike?" The epitome of a lifelike cartoon character, a character that could be a model for the agent of this study, is beautifully illustrated in an encounter with a little boy and his mother that the late Chuck Jones loved to describe [Almeida 2002]. When she introduced Jones as the man who draws Bugs Bunny, her son corrected her, "No, he doesn't. He draws pictures of Bugs Bunny."

Inspired by that earlier example of the illusion of life, we undertook to give the pedagogical agent in the learning environment that serves as the "laboratory" for this thesis an emotional life. To draw students into this fantasy, we wanted a more expressive agent, an agent who seems to be invested in their endeavors and shows some feelings about a struggle and its outcome.

1.1 Pedagogical Agents

[Voerman 1997] claims two particular benefits of animated pedagogical agents: for problem solving, their advice is customized to the student, and for learning enjoyment, they motivate students by making their work more fun.

One recent commentary [Gance 2002], which states that the common coupling of educational software and constructivism is too facile, details four typical elements of constructivist pedagogy: a cognitively engaged learner; hands-on, dialogic interaction with the learning environment; an authentic problem-solving situation; human-to-human interaction (with other learners and with mentors). This last, "social" component has been claimed for human-to-computer interaction [Reeves & Nass 1996], however, and this without an animated agent. If that is accepted, then the others of these four components would be well-served by a lifelike pedagogical agent. Their "presence," that is, their "persona effect" [Lester *et al*. 1997], should help maintain the learner's engagement. And, the problem-solving dialog built into the archetypical virtual design studio satisfies the construction dialog requirement. A "virtual" design studio teaching "authentic" problem solving -- a lot of development effort continues to be invested in this premise, such as underlies the STEVE [Rickel & Johnson 1997] project.

Emotional behavior, if sufficiently believable, could only improve the efficacy of pedagogical agents in constructivist learning environments.

1.2 Thesis Overview

A popular comedian for most of the twentieth century, George Burns, is said to have claimed that "Sincerity is everything: if you can fake that, you've got it made." [Osterweil 2000]

To give artificial feelings to Herman, the animated pedagogical agent that is the denizen of Design-A-Plant (Figure 1.1) and the subject of this research, required an emotions framework and a way of locating him in it; the first was adapted from the work of Ortony [Ortony *et al*. 1988] and a simple way to meet the second requirement was suggested by passages in the dissertation of Ortony's student Clark Elliott [Elliott 1992].

The learning environment that was the subject of and for this work is called Design-A-Plant [Lester *et al*. 1996].

Figure 1.1: Herman The Bug waits as a Design-A-Plant episode begins.

Simply put, DAP teaches middleschoolers how plants will vary in different environments. Taken on an imaginary interplanetary journey, DAP students visit numerous planets with somewhat harsh environments; each environment is a problem to solve by designing a plant with features that mitigate the inherent hazards.

Design-A-Plant and Herman The Bug have been the most successful effort of the IntelliMedia Initiative of NC State, for several reasons. Story and Character are, of course, essential. Beyond those, however, is its learn by doing / building / problem-solving approach to education. And, the multimedia-enriched interface is of paramount importance. The domain knowledge programming also keeps the user engaged.

The story of Design-A-Plant involves companionship, travel, helping, challenges, learning, environmental concerns, encouraging growth.

Herman The Bug is a curiosity: smart, cute, sassy, funny.

Building something is fun. And, it is the essence of constructivist learning, which would say that, in this application, while "constructing" a plant students are also actually building, and/or building onto, their knowledge of plants.

The well developed multimedia content of Design-A-Plant draws the student into the story.

This thesis is organized as follow.

The chapter after this one, **Learning Environment**, will discuss the architecture of the Design-A-Plant software. It will also outline the domain of the learning that DAP offers and describe the challenges this LE presents to students.

In **Behavior Control** (Chapter 3), there will be an overview of the framework determining the emotional state of the pedagogical agent, after first surveying the types of feelings that we wanted Herman to be prepared to convey.

The **An Implemented Agent** section (Chapter 4) will detail how its “Affect Base” is deployed by and evaluated by the LE Controller. This chapter will also describe an example plant-design episode mentored by the emotional Herman.

Evaluation (Chapter 5) will describe informal testing of a prototype at a local middleschool. The Findings of this testing will be presented and discussed.

The next part, **Related Work** (Chapter 6), will present the history of the Design-A-Plant project by reviewing the publications that preceded this one. It will also review other work that algorithmically addresses emotions and that has been reported to produce affect in agents of various kinds.

At the end, (Chapter 7) **Conclusions and Future Work,** will summarize, as well as make some fairly extensive recommendations.

Chapter 2

Learning Environment

The learning environment that is the model for this project is the design studio, common in Design curricula, particularly at the graduate level.

The design studio is especially constructivist, since it typically involves hands-on media manipulation, like painting, sculpture, or prototype mockup. There is an overall subject, such as for a semester's course. The "matter" of that subject is customarily a series of design episodes, which usually start with an objective, something to make or some technique to master. The process of each episode is often iterative, a series of artifact production / refinement steps alternating with critiques, until the objective is met (or, sometimes, until time runs out). At the end of the course, students have learned how to design within constrained environments and, more generally, have learned how to tackle design challenges and/or have developed a library of skills with various media.

And, so it is with a Design-A-Plant session. There is a series of problem-solving episodes. The problem statement is voiced by the pedagogical agent at the start of each episode.

The design session unfolds in front of the student, who makes choices among the artifact component types to assemble their design, which is being critiqued by the agent, who offers encouragement or advice. If the student delays too long while choosing a component, the agent will give advice without prompting.

This advice is programmed to reveal gradually the correct choice(s). First abstract verbal advice is given, then the later, more-actionable suggestions are multimodally delivered.

This learning environment is well-suited to test the efficacy of emotional behavior of pedagogical agents because a learner-tutor relationship is already built-in. Similar to the Knowledge Base, which is consulted by the agent to determine component choice correctness and inform advice assembly, in this research an Affect Base was added in order to enable the agent to express feelings appropriate to the problem-solving context.

2.1 Architecture

A depiction of the parts of the learning environment architecture in this research is shown in Figure 2.1.

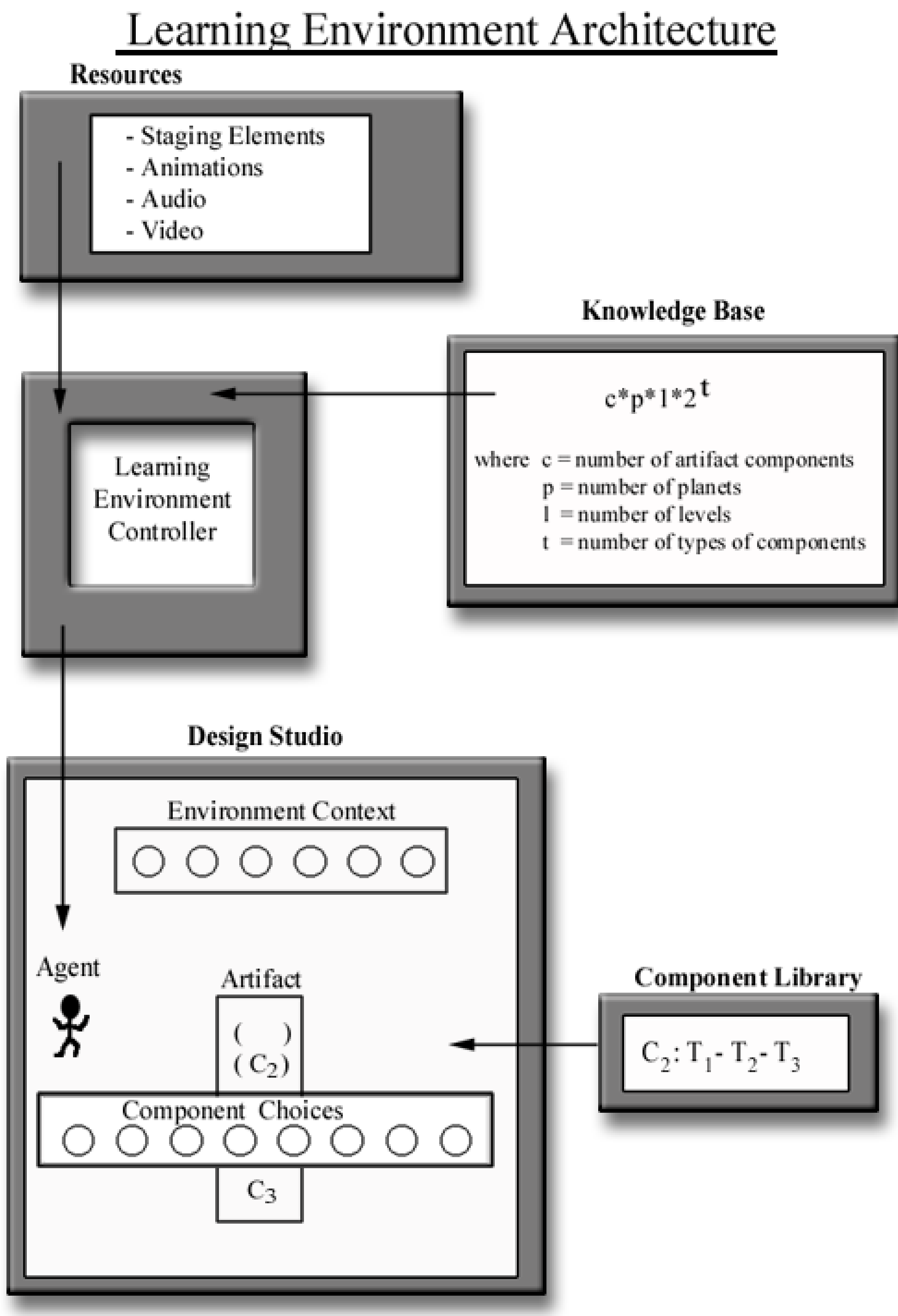

Figure 2.1 Learning Environment Architecture

Design Studio

Essentially, the design studio is what the student interacts with using the mouse, while watching on the screen (Figure 1.1) and listening on their headphones.

Displayed on-screen is the environmental context, which is the combination of one or more elements, often at one or the other extreme, such as rainfall -- high or low -- though some, if present, are only at one extreme, such as nutrients-low. An approximate depiction of this context forms the backdrop of the design studio. This landscape changes among the four types of environments: marsh, desert, cliffs and alpine as the student “travels from planet to planet.”

Also displayed is the artifact under construction, the plant consisting of roots, a stem and leaves. The plant appears in the bell jar shown center-stage in the design studio. Addressing the bell jar, at the level of the plant part being chosen, is the component task bar, which contains eight icons representing the plant-part choices available. As a learner selects one of the icons with a mouse click, their choice is shown “life-size” in the bell jar and connected to any previous choice(s) in the assembling of a complete plant.

Prominently featured is the agent, which at this point in a problem-solving episode would comment on, or reinforce the correctness of, the learner’s choice.

Resources

Among the resources available to the Learning Environment Controller are staging elements, animations, audio and video clips, music.

The staging elements include the graphics of landscapes appropriate to the extremes of the environmental constraints and the bitmaps that show the task bar and its icons which depict the plant-part choices.

The animations are composed of short sequences of stills of the agent in motion, which are terminated with a keyframe[1]. These sequences can then be played in series to result in coherent and believable lifelike behavior. This kind of animation will be used to illustrate Herman's emotions.

The audio and video clips give advice to the student in their problem solving. In addition, some video clips are available to be included as educational "backstory," depicting principles of plant physiology and chemistry, such as transpiration or photosynthesis.

(To clarify our terminology, by "video clip" we mean a "canned" animation, a pre-recorded QuickTime[2] "movie," of Herman demonstrating some problem-solving advice -- for example, encouraging a plant to grow larger leaves so that it can absorb more energy from the sun. We mean to distinguish such "video" from the "animations" that are more

1 A keyframe is one of the frames that make up an animation which is chosen as a reference, such as for timing adjustments or, in our case, to "bookend" movement segments so that they can be dynamically assembled and flow smoothly.

2 QuickTime®, by Apple, is the current standard form of downloadable movie trailers.

like a filmstrip, stored frame-by-frame as Design-A-Plant resources, and played through in real time during an unfolding design episode, to give the agent believable behaviors.)

The music provides an aural backdrop, and its voicing increases with the complexity of the environment.

Following are some details of these resources.

- Animated Transition Behaviors: Played through in realtime, these show Herman moving between keyframes. For example, entering stage left to deliver a narrative about the hazards of a new planet.
- Problem Overview Audios: States the hazards of a new planet by element.
- Believability-enhancing Animations: Herman adjusts to his surroundings or to the flow of the interaction or performs some “fidget,” such as cleaning his glasses or twiddling his “thumbs.” Since these are keyframed, they fit together seamlessly for smooth movement. This type of animation has been adapted to express Herman’s emotions in the implementation of our research.
- Problem-Solving Advisory Audios: These state the environment variable that is hazardous to the plant and identify a dimension of the choice that will fix the problem, such as: “Here we’re building a plant for a shady spot. Focus on choosing between a short stem and a long stem, to allow the leaves to get up into the sunlight for more photosynthesis.”
- Verbal and Visual Transitions: Used to indicate that the advice-giving is multidimensional, such as “and” said with emphasis between two Advisory Verbal Behaviors, or such as Herman moving from a sitting position to reclining before the second-in-a-row of Advisory Animated Behaviors is played.
- Transition Audios: Introduce an Advisory Video.
- Problem-solving Advisory Videos: “Movies” that depict Herman directly solving the plant’s problem with its environment, such as pumping up a too-thin leaf in a low-rainfall environment, or rolling up a leaf that exposes too much surface to the cold in the Alpine environment.
- Direct Suggestion Audios: Usually conveyed as imperatives, these are Herman telling the student what the correct setting is for the dimension(s) of the sub-task that are still not optimal, like “Choose shallow roots.”
- Interjection Audios: A brief word of recognition and/or a random quirky comment.

- Conceptual Explanatory Videos: The educational backstory of Design-A-Plant, these are like Herman's home movies illustrating plant physiology and chemistry. They were not used in our version of the software.

Learning Environment Controller

The LE Controller is in charge of all elements of a Design-A-Plant session. It stages the introductory video, the environment elements descriptive voiceover, the explanation of each problem, the problem-solving episodes and the interspersed transitions. It runs through the animations that make up the various behaviors of the agent and the audio and video clips that are the agent's advice to the student. It consults the knowledge base to determine the correctness of the student's choices, which then informs the advice assembly; and it keeps track of the student's progress.

These are the data used by the LE Controller:

- Environmental Context: The planet type (of the four basic types: marsh, desert, cliffs and alpine) currently being solved.
- Environmental Complexity: A ranking of the degree of difficulty of the four levels of each planet type that would make up the 16-episode grid of a complete Design-A-Plant session. It is roughly based on the total of the planet's sub-task complexities.
- Sub-Task Complexity: The number of component constraints that will need to be satisfied with a sub-task choice.
- Current Complexity: The remaining number of constraint(s) that must be correctly satisfied.
- Multimodal Advisory History: Where the student is in the gradual progression of advice.
- Topic: Which plant part the student is choosing a type of.
- Artifact-based Task Model: Holds the student's choice from among the component instances of the type that is in focus.
- Design Evaluation: The correctness of the student's choice, an encoding of the three-dimensional correctness of each component choice.

- Problem-Solving Idle Time: How long it has been since the last choice made by the student.

Control of the learning experience progresses in the following way:

Session Introduction: The video that introduces Herman and the learner's mission is played full-screen. It ends with a depiction of Herman's (and the learner's) capsule detaching from the starship Green Bean and descending to the first planet that needs plant-design work. A view of this planet then appears as through the capsule's viewport. Herman steps out onto the capsule's dashboard and begins a voiceover that describes the range of environment hazards potentially present in the planets to be visited. Next, on a screen provided center-stage, is played a video that features Herman detailing the variety of choices of roots, stem and leaves available to build plants for these planets.

Episode Introduction: Herman steps out onto the capsule's dashboard and verbalizes a description of the current environment, pointing to the highlighted icon(s) which represents the active hazard(s) that must be mitigated by the appropriate design choices.

Problem solving: The component task bar appears at the corresponding height on either side of the plant-assembly bell jar, after which Herman enters, stage left. As the learner makes choices, advice is given according to the following gradual progression.

1. **Abstract Verbal**

Role: The agent provides the most abstract advice possible, so that 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?’

2. **Abstract Video**

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’.

3. **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.’

4. **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.

Episode Completion: Agent cartwheels across the scene, then does the celebration typical of the current planet, for example, skiing down the mountain in the background of the alpine planet.

Explanation Interjection: Not used in this version of Design-A-Plant, this is the process whereby backstory explanatory videos are assembled together and played, if the controller determines that there is an appropriate opportunity.

Automating Play: Design-A-Plant now plays through automatically. As it times out on each sub-task, it makes the most-wrong plant-part choice, so that Herman's advice thereby teaches all the course content.

2.2 Design-A-Plant

2.2.1 The Domain

There are 16 potential design episodes inherent in Design-A-Plant; however, the most that have been encountered by students during testing is eight.

Planet	Southern marsh	Desert	Tropical cliffs	Alpine
Elements . . . Effects, total Effects, new	high rainfall 1 1	low sunlight 2 2	high temperature 2 2	high watertable 2 2
	high rainfall low sunlight 3 0	low sunlight low watertable 3 1	high temperature low rainfall 6 4	low temperature 6 6
	high rainfall low sunlight high watertable 5 0	low sunlight low watertable high temperature 5 0	high wind 6 6	high watertable low temperature 8 0
	high rainfall low sunlight high temperature low nutrients 7 2	low sunlight high temperature low rainfall 8 0	high wind high rainfall 7 0	low temperature low nutrients 8 0
				--- 79 26

The fundamentals of these hazard - mitigation constraint packets have been researched in NC State's Natural Resources Library [Lange *et al*. 1981, 1982].

low (i.e., minimal) sunlight; 2 packets
In earthly terms, this condition can have two causes. Cloudiness reduces the amount of sunlight or the plant's location is shady. The tendencies of plants in minimal sunlight are to increase their surface area to collect more sunlight and to grow up out of the shade, that is, to lengthen their stem.

high temperature; 2 packets
When a plant is hot it cools off by transpiration, losing water through its stomata. In our terms, it wants to increase its porosity. It wants a stem without bark. It wants leaves that we are calling thin-skinned.

low temperature; 6 packets
Opposite to the effects of high temperature, to survive the cold, a plant wants a stem with bark and leaves that are thick-skinned. It also tries to insulate itself by thickening its parts. And, the plant reduces its exposure to the cold by going for small leaves (i.e., needles, as on evergreens).

heavy rainfall; 1 packet
This is the simplest environment for students to deal with. Here they are taught to go for flexible leaves, ones that don't break off in the heavy rainfall. Herman says the correct choice is thin leaves.

minimal rainfall; 4 packets
Here is another opposite taught by Herman and DAP; thick leaves are a reaction to low rainfall, developed to conserve the water collected by the roots. Similarly, a thick stem is correct. Doing the water gathering, the roots spread out close to the surface, shallow and branching, to absorb the water as it first sinks into the ground.

high watertable (that is, groundwater close to surface); 2 packets
This is a tough environment to solve. The key is that plant roots need to get oxygen. A high watertable can drown a plant. In reaction, a plant will keep its roots shallow, out of the water as much as possible, and will have branching roots, to extract as much oxygen as it can from the water.

low watertable (i.e., low-lying groundwater); 1 packet
Conversely to the previous condition, a plant dealing with a low watertable will extend its roots to take on water. This should be the easier of these opposites for middle-school students to comprehend.

windy; 6 packets
In high-wind environments, plants will tend to reduce their silhouette. They will have smaller leaves. They will grow shorter; in fact, instances of the same species will be shorter on the windward side of a mountain than on the leeward side. They will

strengthen themselves: thicker stems and roots, stems with bark for stiffness, deep roots to hang on.

minimal nutrients; 2 packets
To get its "vitamins and minerals," a plant in soil with low nutrients will tend toward longer roots and will increase its "openness" by developing branching roots for higher absorption of the chemicals it needs.

From these 26 total packets, the 16 different environments inherent in Design-A-Plant are assembled generally in a steadily increasing complexity, while keeping the effects un-confounded. That is, the hazard/mitigation relations never overlap or disagree; there are never two environment factors causing the same effect or conflicting effects.

hazard	mitigation
minimal sunlight	long stem large leaves
high temperature	stem with no bark thin-skinned leaves
low temperature	thick roots thick stem stem with bark small leaves thick leaves thick-skinned leaves
heavy rainfall	thin leaves
minimal rainfall	shallow roots branching roots thick stem thick leaves
high watertable	shallow roots branching root
low watertable	deep roots

windy	deep roots thick roots short stem stem with bark thick stem small leaves
minimal nutrients	deep roots branching roots

2.2.2 The Challenges

The learning environment that was the subject of, and for, this work is called Design-A-Plant.

In story and structure, Design-A-Plant resembles a videogame. That is the point of our work; to provide an environment in which its contemporary audience would be comfortable and ready to take on educational challenges. Indeed, the videogame is a middle ground that DAP's designers, implementers and users all know well.

Design-A-Plant students visit four different planets, as judged by the planets' background imagery and sound effects. However, for each planet, there are four possible levels of increasing difficulty of the design task that these environments comprise. So, there are 16 constructivist learning episodes possible. These environments range from simply rainy, which we say requires only one plant-part type, leaves that are flexible so that they won't be damaged by the rain, to complex environments such as rainy + low-sunlight + high-temperature + low-nutrients, which combination dictates seven (out of the nine possible) specific plant-part types.

More particularly, each plant-building task means choosing a roots system, a stem and a set of leaves. These plant-part sub-tasks involve optimizing three attributes, which can be casually grouped into the part's size, thickness and porosity. For the roots, the size is the depth to which the roots reach, the thickness figures into the roots strength, and their porosity is the branchiness of the roots (more traditionally, their "hairiness"). The stem choice also has a length, thickness for storage or insulation, and its porosity component is the presence or absence of bark on the outside. For leaves, besides thickness, the size is the breadth of the leaves, that is, the amount of surface area they expose to sunlight or to the air; leaves' porosity we express as "skin thickness" in order to speak in the language of our middle-school audience, even though their formal curricula may soon be using such terms as "cuticle thickness" and "permeability."

Chapter 3

Behavior Control

Perhaps the key design objective of our research is to achieve the Illusion of Life [Thomas & Johnson 1981] in the context of learning. Our artists and animators want to put characters on the screen that look real and move believably. Our system designers and programmers have their corresponding intent, code that seems to have intelligence.

Part of the illusion is responsiveness. Does the agent in the software act as if it is aware of your progress? Anticipates the results of your actions? Wants you to succeed?

The overall objective of this work is agent believability. Children are comfortable with pretending and imagining. Therefore, we expected to get a positive response to efforts to make Design-A-Plant's animated agent more lifelike. Further, we would hope this acceptance carries through to the application's educational effectiveness. The intent of the programming of emotion types into the pedagogical agent is to take advantage of expressions of empathy, sympathy and encouragement. Driven by a framework that represents the agent's emotional state, these feelings could highlight problem difficulty, design progress and overall performance.

The specifics of how we set about making Herman more believable follow.

3.1 Emotions

The scheme of emotions used in this extension of Design-A-Plant is a simplified adaptation of that reported in Clark Elliott's Affective Reasoner [Elliott 1992]. Elliott was a student of Ortony, who early on described a way of computerizing the range of human emotions. In fact, the OCC [Ortony *et al*. 1988] scheme has achieved some *de facto* status [Picard 1997] as the scheme of choice for computerizing emotional state.

Of the 24 emotion types described by Elliott (Figure 3.1), we judged 15 to be meaningful in the pedagogical agent - learner interaction exemplified by DAP. These were further reduced to 13 emotion types that could be conveyed effectively with animation + verbal combinations.

Figure 2.2: Emotion types

Group	Specification	Name and Emotion Type
Well-Being	appraisal of a situation as an *event*	**joy**: pleased about an *event* **distress**: displeased about an *event*
Fortunes-of-Others	presumed value of a situation as an *event* affecting another	**happy-for**: pleased about an *event* desirable for another **gloating**: pleased about an *event* undesirable for another **resentment**: displeased about an *event* desirable for another **sorry-for**: displeased about an *event* undesirable for another
Prospect-based	appraisal of a situation as a prospective *event*	**hope**: pleased about a prospective desirable *event* **fear**: displeased about a prospective undesirable *event*
Confirmation	appraisal of a situation as confirming or disconfirming an expectation	**satisfaction**: pleased about a confirmed desirable *event* **relief**: pleased about a disconfirmed undesirable *event* **fears-confirmed**: displeased about a confirmed undesirable *event* **disappointment**: displeased about a disconfirmed desirable *event*
Attribution	appraisal of a situation as an accountable *act* of some agent	**pride**: approving of one's own *act* **admiration**: approving of another's *act* **shame**: disapproving of one's own *act* **reproach**: disapproving of another's *act*
Attraction	appraisal of a situation as containing an attractive or unattractive *object*	**liking**: finding an *object* appealing **disliking**: finding an *object* unappealing
Well-being / Attribution	compound emotions	**gratitude**: admiration + joy **anger**: reproach + distress **gratification**: pride + joy **remorse**: shame + distress
Attraction / Attribution	compound emotion extensions	**love**: admiration + liking **hate**: reproach + disliking

Figure 3.1 Emotion types from Elliott's The Affective Reasoner

The emotion types we employed can be referred to as:

- joy (at meeting a new Student)
- liking or disliking (a new problem-solving episode, based on its complexity)
- hope or fear (toward an episode's sub-task, depending on its complexity)
- satisfaction or disappointment (at the completion of a sub-task begun in a hopeful state)
- relief or fears-confirmed (after a sub-task that had caused a fearful state)
- happy-for or sorry-for (the Student, based on their performance of each episode)
- admiration (of the Student) or shame (by the Agent) (at the end of their session together)

The nine emotion types of Elliott not implemented because they could be out of place in an educational setting encompass:

- distress (as toward a new student whose expectations might not be met -- awaits later DAP version to incorporate a sense of what learners want)
- gloating, resentment, jealousy, envy (following sub-task completion)
- pride (of the Agent) or reproach (toward the student)
- two compound emotions, anger (reproach + distress) and remorse (shame + distress)

- two types, termed compound emotion extensions, love (admiration + liking) and hate (reproach + disliking)

The two remaining (of Elliott's 26) are the compound emotions:

- gratitude (admiration + joy)
- gratification (pride + joy)

These two, while probably useful *per se*, were judged too difficult to depict, especially to middle-schoolers.

3.2 Programming Emotions

As cited earlier, while developing our approach to program emotions into educational software, we studied the Affective Reasoner developed by Clark Elliott [Elliott 1992]. His system allows for modeling the emotional state of interacting "agents," which can be people (such as a Learner) or programs (such as Herman). The Affective Reasoner is a complex representation of human emotionality, in which agents have two-dimensional personalities, one dimension being *disposition* or how they look at the world, and the other being *temperament* or how they express their feelings. As programmed into the Affective Reasoner, emotions are evoked when an agent encounters a "situation," such as an event or an object. By Elliott's "emotion eliciting condition (EEC) rules," agents are affected by a situation to one or more of 24 emotion types (Figure 3.1). Elliott's agents have concern structures, represented as hierarchical frame databases that encode the goals, standards and preferences (GSPs) that make up an agent's personality.

Besides this GSP database, an agent may keep COO (concerns of others) databases about agents they encounter. Although "personality" and "concerns of others" would seem to be compelling directions to take in implementing a pedagogical agent, pragmatically we chose to more narrowly adapt Elliott's work. In Elliott's terminology, as we intended to enhance Herman, he would have a few dispositions and would express each feeling from his limited emotional range with only one temperament. He would be able to express pedagogically useful emotion types, but would remain self-centered; in other words, he would not try to model the student's emotionality.

The process Elliott uses to match frames in an agent's GSP database with its situation results in what he calls the EEC relation. The patterns of this relation, how the situation is construed by the agent, then specify the agent's emotion. Two drawings of this process taken from Elliott's thesis were particularly motivating for our implementation; shown below (Figure 3.2) are the mapping of slots and frames in the agent's database onto an eliciting situation and the EEC relation itself (Figure 3.3). As will be discussed further in the next section, in a manner similar to Elliott's hierarchical frames representation, we used a tree structure (Figure 3.4) to represent the situations that occur during a student's progress through DAP problem solving, and then devised a simple way to construe them as Herman's emotion types.

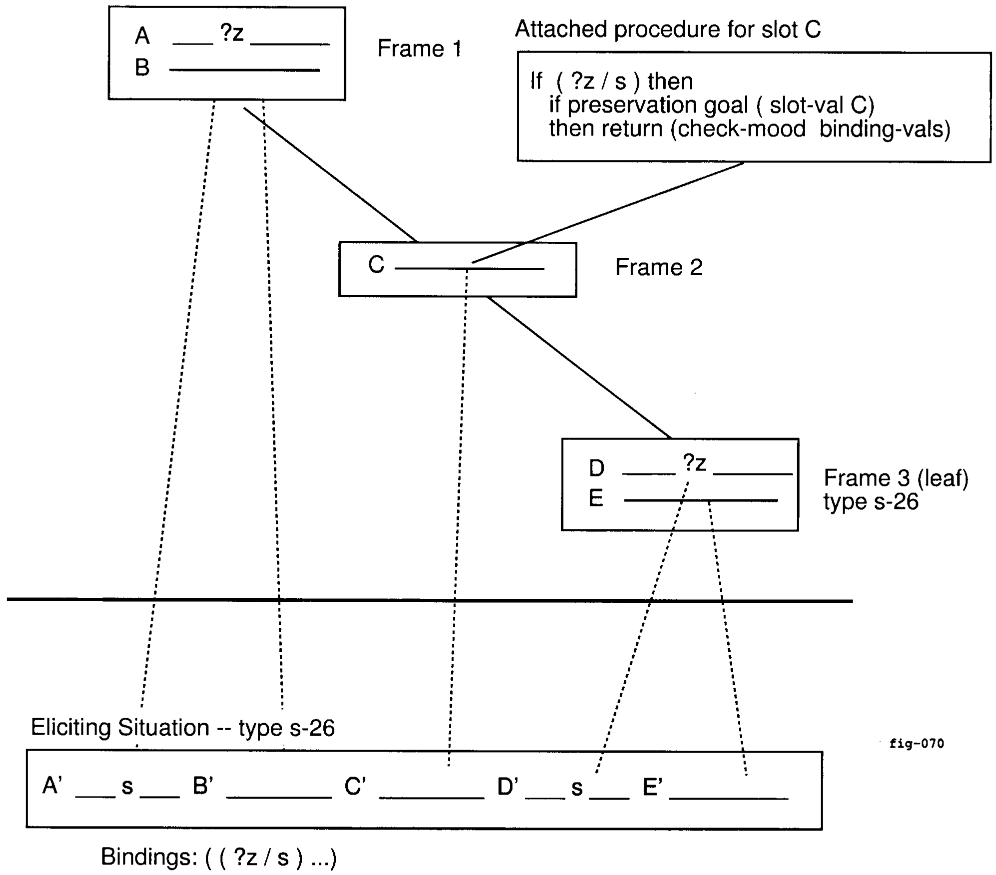

Figure 2.3: Inheritance of slots in a construal frame

Figure 3.2 Inheritance of slots in a construal frame (The Affective Reasoner)

Figure 2.4: The Emotion Eliciting Condition relation

self	other	desire-self	desire-other	pleasing-ness	status	evaluation	responsible agent	appeal
(*)	(*)	(d/u)	(d/u)	(p/d)	(u/c/d)	(p/b)	(*)	(a/u)

Key to attribute values	
abbreviation	**meaning**
*	some agent's name
d/u	desirable or undesirable (event)
p/d	pleased or displeased about another's fortunes (event)
p/b	praiseworthy or blameworthy (act)
a/u	appealing or unappealing (object)
u/c/d	unconfirmed, confirmed or disconfirmed

Figure 3.3 Emotion Eliciting Condition relation (The Affective Reasoner)

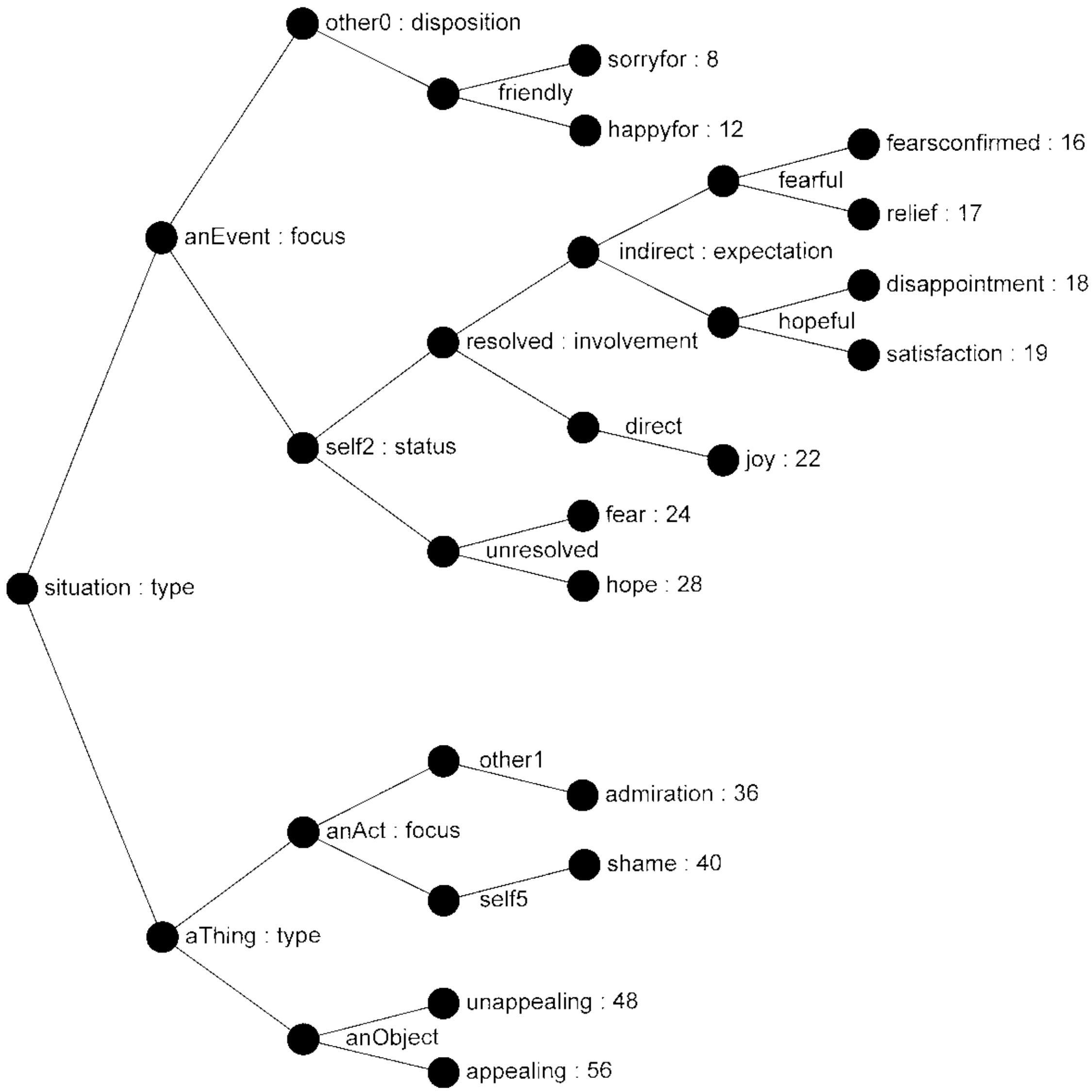

Figure 3.4 Affect Base hierarchy

3.3 Fitting Emotions into DAP architecture

To make the emotional behaviors work, we enhanced some scripts in the Design-A-Plant architecture of Chapter 2 and introduced some additional modules. Resources were added so that Herman can express his feelings. We implemented these feelings with a combination of animation and verbal comment. This combination seems to work. First the LE Controller plays the animation, as described earlier: a series of stills portraying Herman in an appropriate movement. The animation is paused, as it were, at its peak, while the Controller plays an audio that verbalizes Herman's feelings. Then the Controller relaxes Herman's emotional state by playing the animation backwards to the point where he became expressive, such as, back to his impartial-observer pose during problem solving. Examples might be his "hope" emotion, where Herman moves from being an impartial observer to an observer with fingers crossed on both hands,[3] and then we hear him explain why he is optimistic. Or his "fear" feeling, where he turns to face us with an "Oh!" mouth, and then states his concern about the complexity of the current problem solving.

An inherent drawback to programmed and pre-recorded behaviors is that eventually, if not sooner, the agent becomes noticeably repetitive. Natural language generation would help, but that is the subject of other research. Since we are still in "proof-of-concept" territory, however, we chose to minimize the repetition by skipping any emotional behavior that was just previously expressed. Also, the way a DAP session

---

[3] If Herman even has "hands" or "fingers." Anyway, what is at the end of his "arms" has three digits, this we know from careful observation, and he can cross two of them.

decomposes into problem-solving episodes tends to make repetition less noticeable.

The Design-A-Plant fantasy dictates that Herman sit down for the Learner's leaves sub-task -- after all, he is in danger of bumping his head on the top of the screen when the component task bar moves to the top of the plant bell jar! Because he is seated on the task bar, Herman cannot resort to the body language included in his emotional expressions, so he is left with verbalizing any feelings about the leaves' complexity and about the outcome of this sub-task. Besides this fortuitous fantasy boundary, we added to the LE Controller for emotional Herman the requirement not to repeat a problem-solving reaction when moving to the next sub-task in an episode. These two sensitivities together mean that in a given planet visit, if the environment hazards dictate a complex roots decision and the stem choice is then also complex, the Learner will see the expression of Herman's "fear" only once. If the leaves' choice is also complex, they will hear him worrying. We are trying not to dull the effect of Herman's affect.

The major enhancement for emotional behavior is what I've dubbed the "Affect Base." In parallel with the Knowledge Base of Design-A-Plant, which lets Herman grade a Learner's choice of plant part, the Affect Base is consulted to determine how Herman feels during problem solving and at other junctures of a DAP session.

Like our own emotional life, Herman's is something of a "black box," and an object-oriented approach seemed appropriate. The Affect Base generates emotional behavior in response to the following series of commands:

- **startAffect:** The script that sets up the emotions hierarchy, and initializes the totals used to index the case structure that runs the behaviors expressing Herman's emotions.

- **setHierarchy:** Deploys the nodes that represent the Learner's progress through, and the choice points of, a DAP session.

- **setChecked anEvent / resetChecked aThing** (for example)**:** How tree branches are switched on and off so that Herman's emotional state follows the Learner.

- **reportSlot:** The method that, in a switched-on hierarchy node, adds its level's contribution to the hierarchy evaluation total and passes the method on to its ancestor -- an "off" node does not pass on the call and zeros the total. In the hierarchy root, this method stores the total aside so that compound emotions are possible.

- **emote:** This command polls the leaves of the Affect Base hierarchy to identify the emotion(s) that are real in a given situation.

- **behavior "express":** A case structure in which the emotion is expressed that is keyed to the total from the completed hierarchy evaluation.

There is an add-in that demonstrates what might be done to give Herman a slight disposition, that is an interpretation of the Learner's condition. Anticipating the scenario to be described in Chapter 4, here we will mention how Herman is programmed to "scold" a student. If a student makes an apparently uninformed choice -- when they probably haven't weighed all the possibilities, as judged by their not looking at all of the plant-part dimensions while mousing over the component icons -- then, after offering any needed advice, Herman says "However, I think you need to explore a little more. Roll your cursor over all the choices to be sure you have thought about all of the eight combinations. You'll have enough time."

emotional behaviors of Herman by situation[4]:

(new user: joy = Herman does a touch-down! gesture.)
Hi! I'm glad you're going to use Design-A-Plant. You can help me put healthy vegetation on some new planets.

- And, I see that you would like to learn about plants. I like talking about them. So, from time to time, I'll interrupt our work to teach you some plant science.
- And, I see that you like to get high scores. I'll be right here to give you just the advice you need to do that.
- And, I see that you want to have fun while you learn. Most students like my sense of humor -- although it's mostly jokes from plant science. Like, What animals can jump higher than a plant? -- All of them … (plants can't jump)!
- And, I see you want to get through this as quickly as possible, so I'll avoid taking a lot of your time for my corny jokes or for lecturing about how plants work.

If you would like to change your mind on what you want, you can just click on the CHANGE button. (However, here at first, let's visit a few planets, before we make any changes. Maybe you'll be pleased with the way things go.) I'll remind you again later, I promise.

(new user: distress)
- However, I can't promise you that what we have to do will always just be fun; some of it will require serious thinking and remembering and reviewing. Plus, I'll take time to talk about some plant-science stuff, too -- only after things are going well.
- However, if we make this too quick, it won't be EDUCATION!

(new planet)
OK, this is when you can make a change in what you want out of Design-A-Plant. Or, whenever we arrive at a new planet like this. Just click on the CHANGE button when you want to.

(new planet: like = Herman gives a thumb-up.)
Hooray! I hope all the planets we visit will be as friendly as this one. The environment here is relatively easy to build a plant for.

(new planet: dislike = Herman gives a thumb-down.)
Oh my! Here's where we show how well we've learned to build a plant. This is a world with some tough problems. But, I'm really glad to have your help.

---

[4] **Herman's back!** or **Whose side are you on, anyway?**
From my earliest work with agent-inhabited interfaces, I have maintained that, in order to be accepted as well-meaning and helpful, an agent should actually minimize his face time during problem solving. That is, with his back turned and off to the side, Herman is "on our side;" he's not preaching to us or teaching at us.

(new sub-task: hope = Herman crosses his fingers on both hands.)
- Checking my records, I see that students usually do well on this part. And, I'll help when I can.
- Well, although this often gives students some trouble, you've been doing better than average, so I'm not worried.

(new sub-task: fear = Herman faces us with a rounded mouth.)
- We've been having some trouble with complicated situations, so I'm a little worried here. No pressure, though, 'cause I'll have advice when you need it.
- According to my records, this is an especially hard choice for students. I'll try to help when I can.

(sub-task choice)
- Good choice! (Though, actually, as I said earlier, any of the roots would work here.)
- Here, you can just go ahead and pick any of these roots (stems) (leaves).
- (So far,) Not a bad looking plant!
- Good try!
- However, I think you need to explore a little more. Roll your cursor over all the choices to be sure you have thought about all of the eight combinations. You'll have enough time.
- (But,) Nice job of looking over all the choices you have.
- Meanwhile, let me give you some advice about what you just picked.
- Well, now, I took away your cursor so you'd be more likely to pay attention when I'm trying to teach you something. I'm trying not to bore you with too many details or get you upset by repeating things that you already know. So, when you're mousing around like that, it just slows me down.
- Wow, you're fast with that mouse button. Faster than most students are to make a choice. Are you sure you've thought over all the possibilities? Take enough time to make a good decision.
- However, so far you've been doing well. So, I can't scold you too much!
- So many mouse clicks! Maybe you're bored or maybe you're upset with your old buggy buddy Herman.
- However, so far you've been right. So, I can't scold you too much!
- Hang in there, we're making good progress.
- Hang in there, some students just need more time to catch on. This stuff will fall into place for you soon.
- Well, now, that's just where you clicked last time. So, I'm confused. Maybe you think I'll change my mind and take that answer the second time. Not! Or, maybe you've lost interest in this stuff. Or, maybe you don't think my advice will help. Whatever, try to see all the possibilities.
- That choice didn't work for you before in a similar situation. But, of course, I don't mind repeating my advice -- I hope you don't mind hearing it again.

(sub-task end: satisfaction = Herman gives a big Ahah! nod.)
I guessed that picking this part of your plant would be relatively easy, so I'm glad to see I was right! You can do this stuff!

(sub-task end: disappointment = Herman does a heavy sigh.)
Well, I hoped that we could do better with this one. They're only going to get harder. I'm just going by what I know when I give you advice, you'll have to try to catch all the meaning.

(sub-task end: fears-confirmed = Herman shows a grim look.)
I was afraid of this. There was too much to deal with all at once for this part of your plant, but, we'll do much better with this environment the next time we see it.

(sub-task end: relief = Herman has a Mona Lisa smile.)
Whew! This was going to be a tough one, but with your persistence we squeaked by with a win.

(DONE (with a planet): happy-for = An over-the-shoulder grin and thumbs-up.)
0 Hey, good for you!
01 Picking roots
21 on the first try
16 2 out of 3
11 all right
02 Finding the right stem
22 on the second try
14 only missed 1
15 got 1 right
03 Choosing leaves
10 all right the first time

(DONE (with a planet): sorry-for = Herman's shoulders are slumped.)
Well, like I said, this is a tough planet. But, …
I'll keep trying to help you. And, these environment elements will be easier when we see them again on other planets.

(Session / User end: shame = Herman turns to look at us and he's tearing up.)
Gee, I guess I could have been more help with our work. But, I hope you enjoyed Design-A-Plant and learned some interesting things about plants. Bye!

(Session / User end: admiration = Herman blows us a kiss.)
- You're my hero! So many choices and you managed to keep them all in your head. I'm glad to have had your help. Bye!
- You're in the top 25% of the students I've worked with. (You're in the top 50% of the students I've worked with.) (You're in the top 75% of the students I've worked with.) I

hope you will remember some of the things I tried to teach you, and that you will remember me too. Bye!
- You got your wish; Design-A-Plant took you less than average time to complete. I hope you think it was time well spent! Bye!

Chapter 4

An Implemented Agent

Design-A-Plant is implemented with Macromedia's Director® multimedia authoring application. This application, first developed for the Amiga® platform and soon ported to Macintosh® computers, then later to Windows®, has been used extensively to develop computer games, CBT, kiosks, presentations and even for product delivery, such as in installation CDs. DAP was originally deployed on the Mac and in the later stages of our work we ported it to the Windows operating system.

Director has several dimensions; filmmaking is the typical analogy given to explain how it works. A film is usually made, at least in part, on a stage. Similarly, Director has a "stage"; it also has a "cast," which includes resources of the kinds described earlier. Director applications are programmed using its built-in scripting language called Lingo. (In fact, because of the nearly self-documenting, English-like nature of Lingo, we will include parts of the original scripts to illustrate the programming of this project, adding comments for narrative and explanatory effect.) Another dimension of Director is its "score," which manages the appearances “on stage” of the various media elements. A finished application authored with Director either has a .mov extension, for playing within the parent application, or can be delivered with a .exe extension, for free-standing titles.

4.1 Example Design-A-Plant Session

(episode introduction)

Herman pops out onto the left side of the console with an open, somewhat bemused look, then turns to point at the overhead display of environment elements. An audio clip of him plays:

"Look at that, a high watertable. Well, in a marsh that's not such an odd thing. The high rain and low sunlight are also to be expected. Choose the best roots to keep the plant from drowning in the high watertable. And, you already know how to deal with the heavy rain and the low sunlight."

Herman then goes to the left to leave but has a second thought, which he expresses with a thumbs-down gesture:

"Oh my! Here's where we show how well we've learned to build a plant. This is a world with some tough problems. But, I'm really glad to have your help."

Figure 4.1: Herman doesn't like this new planet.

(problem-solving begins)

Herman ducks offstage only to return quickly, turning toward the student with his hands crossed in front of him and an open, rounded mouth. Herman says:

“According to my records, this is an especially hard choice for students. I’ll try to help when I can.”

Figure 4.2: Herman is worried.

The Learner chooses, rather quickly, roots that are non-branching, shallow and thin.

While continuing to stand with his back to us, Herman says:

“The danger from a high watertable is that the plant could drown. As in, ‘agh, I’m drowning, glug, glug, glug.’ It drowns because it can’t get enough oxygen. One thing you need to do is choose between branching and non-branching roots. Which type do you think will spread out to find more oxygen to breathe?”

Then, Herman says:

“However, I think you need to explore a little more. Roll your cursor over all the choices to be sure you have thought about all eight of the combinations. You’ll have enough time.”

The Learner more carefully considers the possibilities (while Herman exhibits no believability behaviors (see description below), because of the sub-task's current difficulty -- in order not to distract the Learner), while the icon pop-ups inform the Learner of the details of each choice of roots. The Learner clicks the icon for non-branching, deep, thick roots.

Herman shows two videos. He sits down before the first one starts and lies down before the second.

The first shows Herman beside a cut-out of a plant's root system, which is just a single taproot. Then the animation shows branches spreading out from the plant's center, while Herman is heard to say:

"Non-branching roots in a high watertable will not be able to absorb enough oxygen for the plant to survive. In a high watertable the roots should be branching so the plant can breathe."

The second video depicts Herman beside a similar cut-out with branching roots that then draw up closer to the surface, while Herman says:

"A plant with dee-eep roots in a hi-igh watertable, will, uh, drown. In a high watertable, the roots should be shallow so that the plant can absorb enough oxygen from the surrounding soil."

Herman sits back up.

The Learner finally chooses branching roots, but ones that are also deep and thin.

Herman is heard to say "Choose shallow roots."

User chooses roots that are branching, shallow and thick.

Herman stands back up and then puts his hand to the top of his slightly bowed head, while an audio of him plays, saying:

"I was afraid of this. There was too much to deal with all at once for this part of your plant, but, we'll do much better with this environment the next time we see it."

Figure 4.3: Herman "was afraid of this."

The component task bar advances to the stem-design position.

With fingers crossed on both hands, Herman expresses some optimism:

"Checking my records, I see that students usually do well on this part. And, I'll help when I can."

Figure 4.4: Herman is hopeful.

The Learner explores the entire range of possibilities, while Herman, now free to do some believability movements because this is a simpler topic, first cleans his glasses, then twiddles his thumbs behind his back, then makes the hoop around his antenna slide up and down (an alien bug's self-absorbed gesture, apparently).

User chooses one of the four short stems on the left side of the task bar.

And Herman is heard to say:

"Here we're building a plant for a shady spot. Focus on choosing between a short stem and a long stem, to allow the leaves to get up into the sunlight for more photosynthesis."

Learner chooses one of the long stems on the right side.

Herman nods his head vigorously, then nearly splits his head open in a wide triumphant, ahah! smile and we hear him saying:

"I guessed that picking this part of your plant would be relatively easy, so I'm glad to see that I was right! You can do this stuff!"

Figure 4.5: Herman's optimism was rewarded.

The component task bar advances to the leaves-design position.

With no room to move at the top of the screen, Herman throws himself to a sitting position on the task bar, and just verbalizes his concern:

"According to my records, this is an especially hard choice for students. I'll try to help when I can."

The Learner chooses leaves that are thin, large and thick-skinned and then hears:

"This is a very rainy environment and the leaves of your plant have to be flexible so they're not damaged by the rainfall. What really matters for the rain is your choice between thick leaves and thin leaves. Which do you think would be more flexible?"

Learner now chooses thick leaves that are small and thin-skinned.

Herman lies down to watch his own self, which is flying around on a jetpack helping a small leaf stretch out to gather in more of the sun's rays, while hearing himself say:

"A plant with small leaves in dim sunlight cannot conduct enough photosynthesis and will have, ugh, no food. We can help this plant by giving it larger leaves. Then it can do more photosynthesis and have plenty of food to eat."

The Learner catches on and again chooses thick leaves, but now they're large.

Herman speaks:

"Whew! This was going to be a tough one, but with your persistence we squeaked by with a win."

Then, Herman looks over his shoulder with a toothy smile and a thumbs-up sign, saying: “Hey, good for you!”

Figure 4.6: Herman gives a Learner well-deserved, positive feedback.

(episode conclusion)

Herman cartwheels across the screen with a “Yippee!” and teleports to the background by the water’s edge where, now wearing a floppy hat, he tosses in a fishing line.

4.2 Learning Environment Implementation

Like the authoring tool we used to develop it, our Design-A-Plant application also has several dimensions. There are four planet types visited in the series of problem-solving episodes that make up a DAP session. There are four levels of environment complexity for each planet type. In each environment, the optimum plant is constructed by choosing one each from eight roots selections, eight stems, and eight leaves selections[5].

These dimensions of DAP are used to specify the cells wherein are stored the *designEval* data for the selections of roots, stem and leaves. That is, a cell is specified by its plant-part type (roots, stem or leaves), the planet currently being visited, the planet's complexity level and the icon position clicked, with the eight icon positions representing binary settings of the three different aspects of the plant parts (reach, capacity, porosity). So, there are

3 x 4 x 4 x 8 = 384 cells

which make up the knowledge base.

As an example of the *designEval* data stored in the knowledge base, here is the roots array:

```
put [
0,0,0,0,0,0,0,0, 2,2,0,0,6,6,4,4, 0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,
0,0,2,2,0,0,2,2, 0,1,0,1,0,1,0,1, 2,2,0,0,6,6,4,4, 0,0,0,0,0,0,0,0,
0,0,2,2,0,0,2,2, 2,3,0,1,6,7,4,5, 0,1,2,3,0,1,2,3, 2,2,0,0,6,6,4,4,
2,2,0,0,6,6,4,4, 0,1,2,3,4,5,6,7, 0,1,2,3,0,1,2,3, 0,0,2,2,4,4,6,6
    ] into rootsEvals
```

These numbers show four patterns. At the simplest level, some of these planets

5 For several problem-solving episodes at the lower levels of difficulty, one or two of the plant parts are not "in focus" and all of their eight types are acceptable. That is, the viability of plant life designed for these simplest environments only depends on, for instance, the leaves chosen -- with any stem or root selection being an acceptable choice -- or on the roots chosen.

have a don't-care pattern of 0s; there is no advice needed about roots for these planets. The next-simplest pattern is made up of two digits, for example the "0,1,0,1,0,1,0,1," and "0,0,2,2,0,0,2,2," subsets above; these indicate a roots system where only one variable needs to be set. Moderately complex environments are shown as patterns containing four digits, as "2,2,0,0,6,6,4,4," and "0,1,2,3,0,1,2,3," for planets whose roots answer means getting two of the three type variables "correct." Finally, there are some planets, like the 14th in the list, where it is cold and there are few nutrients in the soil; for one choice of roots on this planet, advice on all three variables of the roots setting will be needed. This planet's roots encoding is "0,1,2,3,4,5,6,7," with eight digits.

The knowledge base arrays for the stem and leaves choices show similar patterns. Here is how these patterns are derived. The breakdown of the plant part types as illustrated by the corresponding icons and their pop-up / roll-over text is:

leaves

thick | thin

small | large | small | large

thick-skinned | thin-skinned | thick-skinned | thin-skinned | thick-skinned | thin-skinned | thick-skinned | thin-skinned

stem

short | long

bark | no bark | bark | no bark

thick | thin | thick | thin | thick | thin | thick | thin

roots

branching | non-branching

deep | shallow | deep | shallow

thick | thin | thick | thin | thick | thin | thick | thin

A scheme was needed to indicate to Herman when advice should be given and at what level(s), that is, which variable(s) of each triplet should be addressed. Arbitrarily, a

"1," a non-zero value to indicate that Herman should give advice, is encoded wherever that level is "incorrect" in the corresponding sub-task choice.

In the *rootsEvals* list, the roots of the 12^{th} planet are supposed to be branching and shallow. Branchiness is the top level of the roots icons array. The level1 contribution to a particular *designEval* is 0 or 4, depending on whether those roots are branching or not. For Herman to advise the Learner to <u>reconsider</u> roots that are <u>not branching</u>, there must be a 4 in *designEval* for an icon that represents such roots, so the Level 1 contribution is:

0	0	0	0	4	4	4	4

Level 2 of the icons has to encode deep roots as provoking advice from Herman, meaning there is a 2 in *designEval* for those icons:

2	2	0	0	2	2	0	0

Level 3 of the icons is a don't-care, meaning everywhere is 0:

0	0	0	0	0	0	0	0

The complete encoding in the knowledge base for roots in the example episode is the column-wise summation:

2,2,0,0,6,6,4,4

Here is how the instantaneous overlay process works in decoding the particular icon chosen in the Learner's second choice of roots from this example.

The Student chooses icon 5. Retrieving the fifth value in this episode's roots' array above,

designEval = 6

The calculation 6 / 4 = 1 indicates that Level 1 advice is needed.

With ((6 / 2) mod 2) = 1 , Level 2 advice is specified.

From (6 mod 2) = 0 , no Level 3 advice is triggered.

In the example, Herman presents two advisory videos, one encouraging branching roots and the second teaching shallow roots. By our convention, the level-value corresponding to a plant-part type is non-zero when Herman needs to give advice about it. Moreover, this scheme isn't used to encode the "direction" of the advice, that is, toward branchiness or away from branchiness. It merely encodes whether or not to access the advice stored for a particular type of choice in a specific environment. In one episode this plant-part type, such as the "reach" of the roots, might need to be shallow, and in another episode, maybe it should be deep. The same *designEval* Level 1 value of 1 would cause opposite advice in these cases, depending on the specific resource indicated in a list of audios or of video clips.

Another useful list in the knowledge base stores the complexities of the plant-part sub-tasks, such as roots. Again, from our example:

```
put [0, 2, 0, 0, 1, 1, 2, 0, 1, 3, 2, 2, 2, 3, 2, 2] into rootComplexities
```

You can see that the 12th planet in this list has roots of complexity 2, that is, two levels are involved in the choice of icons or, in other words, in the potential advice-giving. As can be inferred, complexity of 0 is a don't-care condition; all the choices are acceptable. Correspondingly, a complexity of 1 describes a sub-task where one level of the three sub-task attributes -- reach, capacity and openness -- is involved; four choices are "correct" and the other four cause Herman to give advice. Finally, for a complexity of 3, only one choice out of eight is correct Sub-task complexity is used to discourage

Herman from performing any of his relatively distracting behaviors. If the current sub-task's complexity is 1 or has been reduced to 1 by the Learner's successful problem-solving, then Herman can fidget, that is, behave believably, while the Learner is deliberating.

When a Learner completes a sub-task after having needed advice no more times than the value of the sub-task's complexity, Herman's feelings turn positive. If he was "fearful" as the sub-task began, his worry is "disconfirmed." If he was "hopeful," that feeling is "confirmed." However, if more advice was needed to complete the sub-task, Herman expresses the negative counterparts of these feelings.

The total of the sub-tasks' complexities for a planet just encountered is used to motivate Herman's thumbs-up or thumbs-down expression of his outlook, after he has described the environments' hazards. Later, Herman's positive or negative feelings when the planet's problem-solving is concluded depend on comparing the total number of advice-givings for the three sub-tasks to the planet's total complexity.

Here are some other details of the knowledge base.

```
put [2,3,4,7,1,3,3,1,6,3,2,3,1,3,1,2] into HermanSaysRoots
put [84,81,83,80,82,79,78,132] into rootsCast
```

The lists like these allow Herman to have the last word if a student is completely lost, so lost that they exhaust their turns at problem solving while continuing to propose non-optimal choices. The value in the *HermanSaysRoots* list for each possible planet represents a/the correct choice of roots. Then, by accessing the *rootsCast* list with this value, an icon depicting this choice is displayed, as if chosen by Herman, who then is heard to say something similar to:

"Ooh, ooh. Wait! I know -- let me get this one."

There is a type of list that allows Herman to keep Design-A-Plant moving ahead if a student is stymied or loses interest.

```
put [2,3,6,7,3,6,6,1,3,3,3,6,1,3,1,8] into rootsMostWrongList
```

The *rootsMostWrongList* holds a choice for each planet visited, which substituted for the student's choice will cause the most advice to be given. In this way, as student could be put back on track.

4.3 Pedagogical Agent Implementation

There are 4 parts to the emotional control process. The first part is deployment of the hierarchy that encodes the slots as outlined by Elliott. The second part is the node(s) switching that occurs as a design episode transpires. The third part is the evaluation of the hierarchy to specify the emotion to be displayed. The fourth part is the expression of the emotion.

Here is the code that sets up the hierarchy to store Herman’s emotional state.

```
on setHierarchy
  set situation to new (script "Root", 6, "Situation")
  set anEvent to new (script "Node", situation, [other0, self2], 0, 1, "anEvent")
  set other0 to new (script "Node", anEvent, [friendly], 0, 2, "other0")
  set friendly to new (script "Node", other0, [sorryfor, happyfor], 1, 3, "friendly")
  set sorryfor to new (script "Node", friendly, [], 0, 4, "sorryfor")
  set happyfor to new (script "Node", friendly, [], 1, 4, "happyfor")
  set self2 to new (script "Node", anEvent, [resolved, unresolved], 1, 2, "self2")
  set resolved to new (script "Node", self2, [indirect, direct], 0, 3, "resolved")
  set indirect to new (script "Node", resolved, [fearful, hopeful], 0, 4, "indirect")
  set fearful to new (script "Node", indirect, [fearsconfirmed, relief], 0, 5, "fearful")
  set fearsconfirmed to new (script "Node", fearful, [], 0, 6, "fearsconfirmed")
  set relief to new (script "Node", fearful, [], 1, 6, "relief")
  set hopeful to new (script "Node", indirect, [disappointment, satisfaction], 1, 5, "hopeful")
  set disappointment to new (script "Node", hopeful, [], 0, 6, "disappointment")
  set satisfaction to new (script "Node", hopeful, [], 1, 6, "satisfaction")
  set direct to new (script "Node", resolved, [joy], 1, 4, "direct")
  set joy to new (script "Node", direct, [], 1, 5, "joy")
  set unresolved to new (script "Node", self2, [fear, hope], 1, 3, "unresolved")
  set fear to new (script "Node", unresolved, [], 0, 4, "fear")
  set hope to new (script "Node", unresolved, [], 1, 4, "hope")
  set aThing to new (script "Node", situation, [anAct, anObject], 1, 1, "aThing")
  set anAct to new (script "Node", aThing, [other1, self5], 0, 2, "anAct")
  set other1 to new (script "Node", anAct, [admiration], 0, 3, "other1")
  set admiration to new (script "Node", other1, [], 1, 4, "admiration")
  set self5 to new (script "Node", anAct, [shame], 1, 3, "self5")
  set shame to new (script "Node", self5, [], 0, 4, "shame")
  set anObject to new (script "Node", aThing, [unappealing, appealing], 1, 2, "anObject")
  set unappealing to new (script "Node", anObject, [], 0, 3, "unappealing")
  set appealing to new (script "Node", anObject, [], 1, 3, "appealing")
  set NodesList to¬
 [anEvent, aThing, other0, self2, anAct, AnObject,¬
 friendly, resolved, unresolved, other1, self5,¬
 unappealing, appealing,¬
 sorryfor, happyfor, indirect, direct,¬
 fear, hope, admiration,¬
 shame, fearful, hopeful,¬
 joy, fearsconfirmed, relief, disappointment,¬
 satisfaction]
end
```

With this code, a network of nodes is created, the Affect Base tree shown earlier (Figure 3.4), which holds the details of how Herman is affected during the Learner’s progression through Design-A-Plant.

The mechanisms to make the nodes work are suggested in [Small 1998], a Lingo scripting treatise, which promotes developing with an object-oriented approach. The emotion types of Elliott became the leaves of a tree structure and the nodes above them represented, more-or-less, the features comprising his Emotion Eliciting Condition relation. Each node has a value (2 raised to the power of its level, which is reckoned down from the root at level 6); if it has been turned on, a node's **reportSlot** method contributes this value to the emotion total and then invokes the same method in its ancestor. A node/feature gets turned on or off as Herman encounters and assesses the various situations in a DAP session. A node that is off zeros the total (and does not contact its ancestor).

Here is the scripting for the root (**630**) and the nodes (**631**) (compare the calls to these scripts made by the **setHierarchy** script above).

```
630
-- root script
global ETotal, affect
property myDepth,myName

on new me,aDepth,aName
  set myDepth to aDepth
  set myName to aName
  return me
end

on reportSlot me
  put ETotal into affect
end

631
-- nodes script
global ETotal, situation
property ancestor,descendants,originalAncestor,myBit,myLevel,myChecked,myName

on new me,ancestorGiven,descendantsList,aBit,aDatum,aName
  set ancestor to ancestorGiven
```

```
  set descendants to descendantsList
  set myBit to aBit
  set myLevel to aDatum
  set myName to aName
  resetChecked
  return me
end

on reportSlot me
  if myChecked then
    set myPower to (the myDepth of situation - myLevel)
    set myPart to myBit * power (2, myPower)
    set ETotal to (ETotal + integer(myPart))
    passUp
  else
    set ETotal to 0
  end if
end

on setChecked
  set myChecked to TRUE
end

on passUp
  reportSlot ancestor
end

on resetChecked
  set myChecked to FALSE
end
```

Here are code snippets, used for a situation where a new sub-task is encountered:

```
  checkEvent
  checkSelf
  checkUnresolved
  if (currentComplexity > 0) then
    set AllSettingsWork to FALSE
    if (currentComplexity < 2) then
      checkOptimistic
    else
      checkPessimistic
    end if
    emote
```

```
    if Affective then
      -- dealing with concern about repeating last sub-task emotion
      if affect <> prevAffectBefore then
        set prevAffectBefore to affect
        behavior "express"
      else
        nothing
      end if
    else
      nothing
    end if
  else
    set AllSettingsWork to TRUE
  end if

on checkPessimistic
  setChecked fear
  setChecked fearful
  resetChecked hope
  resetChecked hopeful
end

on emote
  set ETotal to 0
  set affect to 0
  reportslot sorryfor
  reportslot happyfor
  reportslot relief
  reportslot disappointment
  reportslot fearsconfirmed
  reportslot satisfaction
  reportslot joy
  reportslot fear
  reportslot hope
  reportslot admiration
```

```
  reportslot shame
  reportslot unappealing
  reportslot appealing
end

on behavior action
if (action = "express") then
   if canMove then
     case affect of
     .
     .
     .
24:play_fear_forward
        repeat while (current_frame <> end_frame)
          behavior "tick"
          updateStage
        end repeat
        sound playFile 2, AUPath & "fear"
        repeat while soundBusy(2)
        end repeat
        play_fear_backward
        repeat while (current_frame <> start_frame)
          behavior "tick"
          updateStage
        end repeat
```

The **emote** procedure polls each leaf, invoking its **reportSlot** method. If that node represents an/the valid emotion type, the message chain extends up to the root, which stores the *Etotal* value in a second variable, *affect*. (There is a second variable, and the polling process is not stopped when the root is reached, because of the potential for compound emotions in later versions of our code.) The variable *affect* is used as the index in the **behavior "express"** case statement of emotional expressions.

Chapter 5

Evaluation

5.1 Context

We conducted an empirical evaluation at Ligon GT Magnet Middle School in Raleigh. This time we found our student testers in Ms. Thompson's technology lab; we also tapped students from across the hall, Mr. Thibault's classes (his "morning news" video journalists, for example). We were onsite at Ligon Friday, December 4, 1998 through Wednesday, the 9^{th}, working with five or six students per day, each in one class-period of about 45 minutes, to experience Design-A-Plant and tell us about it.

Most of the students were in seventh grade. They brought signed permission slips. They worked one-at-a-time at a computer we had brought to their lab, the same computer with which our version of Design-A-Plant had been developed. We set them up to use headphones to minimize distractions. Because we ran the full intro system (including

Herman's "Hello, Recruit" animation, his "These Are Your Environment Elements" voiceover, and the "512 Possible Combinations" plant-part settings overview movie), we did not really offer any introductory remarks, except to tell the students that, with Herman's help, they would design plants for two different environments. These two environments were of roughly similar difficulty, and the advice system worked the same in both. The one with traditional Herman was Planet Seven of the 16 potential episodes in a full DAP, and the one with emotional Herman was Planet Nine. Seven has just two active elements, light rain and high temperature, but six plant-part settings need to be correct -- branching and shallow roots, a thick stem with no bark, thick leaves with thin skin. Nine is nominally harder, with three environment elements active, heavy rain, low sunlight and high watertable. Here Herman is free to express his feelings, and on-sight, he calls Nine a "world with some tough problems." The ideal plant for Planet Nine has branching and shallow roots, a long stem, topped off with large but thin leaves.

The programming alternated the order in which the two environments were presented to the students; the one with emotionality was always the same Nine, but if it came first, then for the next student it was the second environment seen (and vice-versa for Seven). After the two environments were solved in succession, the program closed and we interviewed each student briefly before their class-period ended.

We asked them what difference(s) they noticed between the two planets, what difference(s) in Herman's behavior. If they described a difference, we asked which Herman they preferred. When we were finished with that topic, we asked each of them to tell us how they relate to educational software like DAP; among these types of students, would they say that they are the kind who:

- likes to play games.
- wants to be entertained.
- is interested in learning (the specific material).
- likes to solve puzzles.
- wants to get high scores.

This survey was to inform some later effort at user modeling. If there was time left, we asked the students for general comments about Design-A-Plant and for suggested improvements.

5.2 Findings

First, however, some disclaimers. This evaluation was informal, involving unscreened subjects, using a non-rigorous protocol, and the planets comparison could be biased, i.e., just the order of the two was swapped between successive students, not the character of the planets -- the same one always had the emotional Herman. The evaluation was limited; there was one instance of each kind of agent and there was no control of exposure to them -- students might have been preconditioned to spend less time on one or the other planet and, so, not get the full effect of the difference. The evaluation was simple; no attempt was made to gather any performance data while the students were solving the two environments.

Then, the findings. Of the 22 students, 15 said that they noticed the difference (or a difference – one student commented that Herman had "more to say" on one planet -- on the affective planet, he does!) between the two environments. They expressed it as:

- Herman was more active.
- Herman was worried for the plant.
- Herman wanted you to get it right.
- Herman was "just acting normal."
- Herman's feedback is a confidence-builder.

Of the fifteen students who noticed a difference, eleven said they had a preference; eight of them liked emotional Herman better and three of them preferred original Herman. The students explanations of their preference were like these:

- It's helpful to know if the sub-task is hard or easy ahead of time – you know what to concentrate on.
- His feelings were recognizable.
- He explained better and understood that some of the sub-tasks were hard (but unemotional Herman would be better for education software because he gave more information)[6].
- He had more helpful things to say, such as telling you that things were hard.

One student reported that when Herman said picking a stem would be easy, she became especially concerned to get that task right (though she didn't claim to notice the difference between Hermans).

[6] As Spock is more logical without emotions?

Chapter 6

Related Work

6.1 Design-A-Plant

"If I have seen further it is by standing upon the shoulders of giants."
Isaac Newton

DAP history

The NC State course-offerings publishing system TRACS listed the following in Spring '94: CSC 591D SPTP-DN MULMED INT LESTER, J.

As it developed, my advisor, James Lester, was part of a "team" teaching a class on multimedia interface design; the rest of the team was Patrick FitzGerald, of NCSU's School of Design. In addition to attracting students from Computer Science and Design, their class also pulled in some from the Mathematics, Psychology and Mechanical Engineering departments. It was a "project" class and the project became Design-A-

Plant. The diverse backgrounds of the students, when mixed with the contagious enthusiasm of the teachers, the interesting challenges of interface design, the familiar/addictive subject matter of the computer "game," and the approachable subject matter of the learning environment, yielded a very fertile nutrient medium. The bumper crop that came up included an engaging opening animation, a multimedia overview of the topic, numerous illustrative videos for use as advice to student travelers, several "backstory" illustrations of interesting botanical mechanisms (e.g., photosynthesis and transpiration), four planetary environments with multivoiced musical themes depicted in a graphical interface that contains the plant design studio and is the classroom for that well-traveled, well-spoken pedagogical agent, Herman the Bug, explorer, lecturer, coach, gymnast, stuntman, lame comedian. All this from a class of about two dozen students, who were more-or-less equally divided between programmers and artists, and between undergraduates and graduate students.

After the ferment and fervor of this class subsided, Lester and FitzGerald set about systematically broadcasting their new AI-Design hybrid, by attracting substantial, initial funding as the IntelliMedia Initiative and by aggressive publication.

The first of the DAP papers, by Lester and the lead programmers, went to the heart of the DAP experience, the design-centered learning environment, and its constructivist foundation. This paper detailed how Design-A-Plant's clever interface design and programming continually focus the learner's efforts to build (upon) their knowledge of how environmental factors influence plant physiology. This focusing is done with a tripartite contextual representation; environmental, artifactual, and advisory indicators follow the student's progress and are used to motivate the system's interventions.

A second paper about DAP programming details how Herman the Bug's behaviors are dynamically assembled to give him three essential attributes: timely domain coverage, contextuality, and continuity. Design-A-Plant's behavior space contains eight families of behaviors and was assembled in three steps; furnishing the space with the behavior types, structuring the space, sequentially navigating it in runtime. The behavior space is structured by ontological, intentional, and rhetorical indexing, with pedagogically appropriate prerequisite ordering, using a visual continuity metric. The authors emphasize several design maxims: persistence (of agent and pedagogical object), immersion (what we have called presence), verbal support (reminders, interjections, bridges), and contextualized music.

Focusing on the interface design part of the Design-A-Plant project, a third paper, by IntelliMedia's two principal investigators and Brian Stone, described it as a "pedagogical design studio." They declared that such an interface is ideal for constructivist learning, especially by children. They say it should be made up of a segmented design task and an artifact-based task model with its own segmentation in parallel, and it should have the look and feel of a design studio. The authors point out that with an artifact-based task model, a student's progression through a multi-level learning environment such as Design-A-Plant can be programmed to adjust to uncommon strengths or weaknesses in their domain knowledge. The agent's advice library and the interface's soundtrack are also motivated by the task model segmentation and tracking.

Next, Lester and Stone published a paper about increasing the believability of their pedagogical agent. This work can be seen as an outgrowth of their early observations of and interviews with students interacting with Design-A-Plant. They were convinced that a more believable pedagogical agent would better motivate the students using their learning environment. However, they describe three concerns: behavior(s) to increase believability should not detract from learning, believability enhancement should complement the agent's advisory and explanatory behaviors, routine behaviors would be detrimental. The way to resolve these concerns is with dynamic animation; situated liveness shows the agent to be alert and aware, controlled visual impact keeps the agent's activity at an amplitude that matches the student's (subtle partial-body movements, during problem solving, when the student is presumably considering their options, contrasted with more demonstrative activities, such as Herman's cartwheeling across the studio when the plant design has been successfully done), complex behavior patterns are unlikely to be dismissed as "programmed."

The authors describe a competition-based sequencing engine that dynamically animates their pedagogical agent. Potential behaviors are periodically, frequently promoted or retarded according to their appropriateness for exhibition, relative to the student's problem-solving activity and to the agent's behavior history. When a behavior is chosen and exhibited, its potential is zeroed out, so that others with slower growth rates then have a chance to be seen. The student gets a pedagogical agent that doesn't seem to be predictable.

Another paper from Lester's group describes Design-A-Plant as an example of mixed-initiative problem-solving. It describes how the software goes about controlling initiative and providing explanations. The prerequisites of learner/agent dialog built into DAP: a testbed learning environment (design-centered), a problem-solving context model (environmental, artifactual, advisory), and the animated pedagogical agent. The requirements for managing the dialog, controlling the initiative, and providing the explanations are a behavior space and a behavior sequencing engine.

Lastly, prior to our undertaking to extend the programming in Design-A-Plant, a large-scale evaluation was done at a local suburban middle school. This was a milestone event, carefully executed; the key ingredients for a definitive study of knowledge-based learning environments with animated pedagogical agents were finally available: a fully functional APA, in an interactive learning environment, for an ideal population of students. The subjects of the evaluation were five agent "clones," which were distinguished, one-from-another, by the way they delivered problem-solving advice to the students: animated (Herman shows a "home movie," demonstrating the principle that connects the environment element(s) and the plant-part variable(s)), verbal (in a voice-over, Herman tells about the principle), direct (Herman says things like "Choose a long stem!"), multimodal (i.e., all of the preceding, in succession, if errors continue), unhelpful ("You better think again!"). Each agent was evaluated by a group of 20 students, half were boys and half girls. The process the students followed was to sign a consent form (personally; the students had already brought in permission slips, but this small step was a way to treat them as equal participants with the graduate students in the software evaluation; a nice touch), then work through a series of pre-DAP assignments. These were a demographics questionnaire, a subject matter multiple-choice test (in fact, as judged by this pre-test, no differences were found among the clones as far as their students' prior experience), a group of eight manual index-card sorting exercises (given these plant parts, separate those which are helpful, to a plant's survival in each environment, from those that are hurtful). The multiple-choice test and the sortings were meant to simulate the problem solving built into the version of Design-A-Plant being tested. Just before DAP itself, there was a training module delivered by a research assistant, carefully and reproducibly describing the challenge, the interface, the agent, the variables. Then, Design-A-Plant, eight planet visits, in the same order for all students of all clones; the early four were simple problem-solving with one environment element, the latter four were complex, with two or more elements active on each planet visited. Then came post-DAP testing, with the same multiple-choice questions (rearranged) and the same card sortings.

The data showed that the students learned (had post-test scores significantly better than their pre-test scores) from the more-expressive agents, the multimodal, the animated, and the verbal; the direct and the nonhelpful agents did not improve test scores significantly.

The authors' conclusions were:

1. The students learned; probably not simply by time-on-task or practice, because learning with the less-expressive agents was not as great; the agent-environment combination was the reason why the students learned.

2. An animated agent in an interactive learning environment makes the students feel good about their experience; this could be because the agent stimulated ongoing reflection or because he motivated; all the agents, even the nonhelpful one, were given positive ratings by the students.

3. The multimodal agent was significantly better received than the others; this could be because the richer delivery: more media types and more advice levels.

The Hermans' (all the clones') positive ratings were probably real because:

- students were treated as anonymous, and they were told that they would be.
- students were left alone while they filled out their reviews.
- students were encouraged to sincerely review the software in order to help improve it.

The author's recommendations:

1. Designers should put animated pedagogical agents into interactive learning environments.

2. If not a animated pedagogical agent, at least an animated agent should be considered.

3. A multimodal, animated pedagogical agent would be especially effective.

As well, in this large-scale evaluation, data were gathered during the problem-solving sessions within Design-A-Plant, and their meaning and significance were reported in a second paper.

Several questions were answered:

1. Just to be sure, were the more-complex problems (environments with multiple elements) harder to solve? Answer: yes.

2. How did the type of agent affect the student's performance;

- for the simpler, one-element problems, no difference.
- for the complex problems, three levels of effectiveness were seen: the best were the multimodal and direct clones, next-most effective were the animated and the verbal agents, least effective was the nonhelpful one; these three groups were statistically distinct.

(We recall that a similar spectrum of improvements had been seen looking at the pre- and post-test results; the multimodal agent produced the greatest improvement and the nonhelpful, the least.)

The authors' findings:

- Animated pedagogical agents improve students' performance.
- The agent that was both multimodal and multileveled was the best.
- The effect of agents is greater for complex problems than for simple.

The authors' conclusions:

- Animated pedagogical agents can improve problem-solving.
- Educational software designers should put in an APA.
- Multimodal agents are critically effective.

6.2 Emotion Programming

In the design phase of this work, we found one recent survey of issues related to affective computing especially inspirational [Picard 1997]. This book is well written and thoughtful, especially concerned with systems that seek to observe and accommodate users' feelings and attitudes. There are other milestone reviews [Pfeifer 1988, Frijda 1986] of AI and emotions.

The computational frameworks for modeling emotions, proposed and implemented, cover a wide range, as wide as the theories of what human emotions are and where they come from.

At one end of the range are the cognitive / motivational approaches, like the OCC model. At the other end are the "architecture-based" / "design-based" approaches, as the Cog/Aff scheme [Sloman 2001] and the semiotics rationale, as for example of [Gonçalves & Gudwin 1999]. Somewhere in this range is the hormonal model [Cañamero 1997] and a fuzzy logic model [El-Nasr & Skubic 1998]. The former has led to use for autonomous agents [Allen 2001]; the latter is based on the Intelligent Agent framework of a standard AI text [Russell & Norvig 1995].

The following are some noteworthy examples taken from this range of theories.

The DFKI group have recently launched their Presence project [André *et al*. 2000], which will model an agent's affective state to motivate conversational dialog with the system's user. They have chosen to work from the OCC model of emotions, which they describe as a "model of causation …to determine the affective state of the character in response to events in the environment." The Presence agent is Cyberella™, who stands to the side of a slideshow, gesturing to parts of the text, and carrying on a conversation with the visitor -- somehow looking even more impressive in German.

An especially appealing use of believable emotional agents is the "intelligent virtual environment (IVE)" of the dolphins Tristão and Isolda [Martinho & Paiva 1999]. These authors report that the personalities of their agents are specified using OCC theory. However, their behavior is controlled by a "character cycle" which is based on the two-stage emotional stimuli appraisal process from Frijda's theory [Frijda 1986]. The four-phase character cycle is perception, reaction, reasoning and action. The configuration objects and production rules of the IVE system are implemented with CLIPS [Giarratano & Riley 1998].

6.3 Affective Pedagogical Agents

Other research in our own group [Lester *et al*. 2000] has been influenced by the OCC computational model and Elliott's Affective Reasoner framework. However, rather than build an elaborate scaffold to reckon the agent's emotional state, these authors have focused on the multimodal communication that is significant to problem-solving in tutorial dialogs. Their agent Cosmo was designed and rendered to be able to convey a range of emotive movements, subtle facial expressions though full-body behaviors, including arm movements and hand gestures. In order to maximize the power of emotive behavior, Cosmo expresses his feelings with animated behavior crafted to complement a range of speech acts: of congratulations, of (rhetorical) questioning, of concern about difficulties and of thoughtfulness. These behavior types and speech-act types are mapped one-to-one and the agent's movements populate an emotive-kinesthetic behavior space to guarantee visual continuity. Additionally, there are “micromovements” that provide transition and “presence.”

Some authors have argued [Brna *et al*. 2001] that building emotional pedagogical agents is a dubious, even dangerous, undertaking (while at the same time describing their own early attempts and future plans). Their domain is multimedia authoring software for young children (5 - 8 years old); T’rrific Tales lets them author cartoons with the help of a ‘nice’ girl agent, Louisa, somewhat older than her students. She appears in a window on their desktop, engaged in her own work, until they request her help with a click on her window. The authors say that Louisa then turns from her work at a graphics tablet and expresses herself in text and speech. They outline six aspects of behavior to be implemented in an agent that represents an empathetic teacher, and they claim that their agent does all but the final aspect. Louisa is *available* to the students. She *attends* to their needs by interrupting her activity. The agent, like a teacher, observably *engages* the student, tuning in to their “actions, thoughts and feelings.” Louisa expresses the *value* in which she holds the student and their work, then tries to *encourage* them to progress. *Parting* is when the agent moves on, to her own work or to help someone else; this is where they acknowledge further work is in order. No details of an emotional behavior framework are included. Instead, the article is concerned with the potential, sincerity and ethics of empathetic agents as teachers’ helpers.

Similarly, preliminary work has been reported that would have been done about the time of our testing [Abou-Jaoude & Frasson 1999]. These authors talk of creating a “believable ITS” (that is, Intelligent Tutoring System), by adding a layer to its programming, so that, in effect, the customary student model is enhanced with an emotions and believability layer. This layer also includes personality traits and "human" randomness. Their approach to emotional programming is based on the OCC framework, which they express as couples of contradictory emotions, such as "hope" and "fear." The value of each of 13 couples varies from -1 to +1. The instant emotional status is computed every time an event takes place, using a matrix whose element weights are experimentally determined in a backtrack procedure with user input -- human input to promote human-like system behavior. Their user model is actually a competitive learning environment incorporating, besides the tutor and the emotional agent, a "troublemaker" to "sometimes mislead the student for pedagogical purposes," in the role of a classmate of the user. This competition leads to another value from users testing, the deception degree, dd, the degree of disappointment that the agent registers in his relationship with the troublemaker. As might be expected, their future plans are similar to ours, intending to compare an ITS imbued with a believable agent to the same system with "no emotional nor believability aspects." However, they are also seeking to validate some interesting preliminary results: "a believable tutor would be very successful with users who have a low to average knowledge level, but less appealing to experts."

Recently, other research groups have recognized the applicability of the OCC model [André *et al*. 2000] and, more to the point, even used it to motivate the behavior of pedagogical agents [Gratch 2000]. This latter system, Émile, deserves further comment. This is a plan-based approach, done in a context mentioned earlier [Rickel & Johnson 1997], utilizing updates to OCC [Reilly 1996, Elliott 1992]. Its author claims that a plan-based approach generalizes the process whereby events are construed, the first step to determining any resultant feelings. Secondly, planning is said to make it easier to convert construal frames to emotion-eliciting conditions. In particular, derivation of the Self, Desire-self, Status and Evaluation conditions is especially facilitated. Five emotion types are implemented in Émile: hope / fear, joy / distress, and anger. The intensity of each of these is determined from its importance and probability. The approach of this author would seem to lend itself well to an ITS with a well-developed student model that can then inform planning on behalf of the tutee, so that mutual goals can be met. Also, the author says that other techniques developed for plan representations could enhance Émile, like Cooperative Plan Identification [Young 1999] to provide plan descriptions.

Another publication [Marsella *et al*. 2000] from the same research group is based on the emotion theory of [Smith & Lazarus 1990] because, they say, of its emphasis on stress and coping. Their application is also a multimedia title, designed to teach problem-solving skills to mothers of children with cancer. The authors describe a two-step appraisal process. First an event's relevancy to, and coincidence with, an agent's motivations is determined. Then, secondary appraisal looks at four factors: accountability, expectancy and two types of coping potential, problem-directed (adjusting the agent's situation) and emotion-directed (adjusting the agent's psychology). This appraisal process sounds similar to the OCC framework. With its emphasis on story, pedagogical goals, cinematography and character, this project could be a highly effective platform with an intrinsically rewarding use of affective programming.

Chapter 7

Conclusions and Future Work

Education moves with the times and it is always challenging. Teaching means spending “quality” time in the classroom and more time outside to be ready for the classroom. Learning carries similar responsibilities. Fortunately, there are people who enjoy either and/or both sides of the desk. And, there are people who can make either and/or both enjoyable.

Personal computing can mean empowerment to students and fulfillment to teachers. It can meet the former where they are today: on the edge of edutainment. The latter can now convey concepts with simulated problem solving and manage to teach them meaningfully, enjoyably and enduringly. With well-designed graphical user interfaces and powerful multimedia processing, students’ attention can be held and renewed, as they interact and progress.

One recent addition to the personal computing interface is animated pedagogical agents. Often purposely modeled after cartoon characters to encourage suspension of disbelief, these creatures can inhabit simulations to focus users' attention and to maintain the pace of problem solving, such as in constructivist learning, where companionship of learner and mentor-expert is a key ingredient. The animated pedagogical agent can give contextualized advice during problem solving but will lack the human touch, which might be conveyed with emotional responses, indicative of concern or encouragement or satisfaction at well-earned progress.

An affective pedagogical agent, that is an agent which expresses recognizable emotions, could reinforce the learner's suspension of disbelief. Such an agent could "know" how hard the current problem being solved is, for this student, if they've worked together before, or for most students, if the agent has been keeping a journal, and as the learner makes progress. Such an agent can "disarm" a student, if it infers their lack of attention or their frustration, or if it notes they're not "getting it," despite good advice.

7.1 Future Work

- The obvious extension of our preliminary evaluation is a comparison of the pedagogical agent with and without the emotional behaviors. There should probably be a third version included, one which unemotionally conveys the information included with the feelings expressed. In the example, we have Herman saying, before the difficult roots task, "According to my records, this is an especially hard choice for students. I'll try to help when I can." Afterward, when his fears have been confirmed, he says, “I was afraid of this. There was too much to deal with all at once for this part of your plant, but, we’ll do much better with this environment the next time we see it.” In this third version, we would want Herman to comment, but matter-of-factly, about the outlook and the outcome. These three versions could then be compared with respect to the understanding gained by Learners, with pre- and post-testing of the constraint packets in the knowledge base.

- Positive feedback should be included in Herman’s comments, so as to reinforce what the Learner got right, or to teach what they guessed well.

- We should gather and use any available interface events. For example, we might make these kinds of assumptions:

pointless mouse movements (while they would have no effect)	=	not paying attention
making task choice “too soon”	=	not thinking / not caring
repeating (same previous) choice	=	not engaged
clicking madly	=	frustrated / bored / upset

and have one or more standard “intervention” behaviors for Herman, to demonstrate his awareness and concern.

- There is room for some subtle, almost cosmetic, improvements. One would address the arrangements of the icons representing the plant-part types in each sub-task. It would probably improve learning efficiently if these attributes were more parallel, that is, “reach” of the plant parts, followed by “capacity” and, at the lowest / finest level, “openness,” would be the order for each topic. Compared to the current breakdown of the plant-part types shown earlier, a more-parallel arrangement of the icons and their pop-up / roll-over text would be:

leaves

- small
  - thin: thin-skinned, thick-skinned
  - thick: thin-skinned, thick-skinned
- large
  - thin: thin-skinned, thick-skinned
  - thick: thin-skinned, thick-skinned

stem

- short
  - thin: no bark, bark
  - thick: no bark, bark
- long
  - thin: no bark, bark
  - thick: no bark, bark

roots

- shallow
  - thin: branching, non-branching
  - thick: branching, non-branching
- deep
  - thin: branching, non-branching
  - thick: branching, non-branching

- Another subtle, cosmetic improvement would be to increase the granularity of the progression of hinting. In the example, the optimal roots choice is branching and shallow. The first time, the Learner chooses non-branching and shallow. Herman says, “The danger from a high watertable is that the plant could drown. As in, ‘agh, I’m drowning, glug, glug, glug.’ It drowns because it can’t get enough oxygen. One thing you need to do is choose between branching and non-branching roots. Which type do you think will spread out to find more oxygen to breathe?" On their second try, the Learner chooses roots that are non-branching and deep. Herman shows two videos, one depicting the advantage of branching roots to increase oxygen absorption and the second showing that shallow roots will keep the plant from "drowning." Since this was just the first mistake on roots depth, the hint might more properly be the audio, hazard and variables, abstract discussion, saving the more-direct video until later.

7.2 Concluding Remarks

This thesis reports that we have made a pedagogical agent more lifelike by giving it affective behavior. We were able to do this by addressing the so-called "cognitive" emotions, that is, emotions of a high enough level for us to be aware of them -- internally and in others. We trimmed the traditional lists down to feelings which are appropriate to the situations that arise during expert-assisted problem solving and, as well, would be positively viewed by the learner. We devised a hierarchical framework to store the agent's condition in these situations and to evaluate it on the "spur of the moment," as at the conclusion of a successful but involved problem-solving session. The agent's expressibility was greatly enhanced by matching his usual spirited verbalizing with clever animation of his facial expressions and body language.

We enlisted 22 middle school students to work with the agent, in problem-solving sessions, first the un-emotional version, then the affective one (or vice-versa). The students, if they had a preference (which 11 did), liked the emotional agent better (by 8 to 3). Several of the students said they liked him to alert them when the problem solving was hard, and one said that if the agent said the task was relatively easy, they sure wanted to be right!

We think our efforts to improve Design-A-Plant by giving Herman feelings are quite promising. In our evaluation, the students got the difference, they liked it and responded to it. We got the encouragement we hoped for. Affective Herman was seen positively -- helpful, caring, motivating.

Bibliography

[Abou-Jaoude & Frasson 1999] Abou-Jaoude, S., Frasson, C. (1999). Integrating a Believable Layer into Traditional ITS. *AIED'99: 9th International Conference on Artificial Intelligence in Education*.

[Allen 2001] Allen, S.R. (2001). *Concern Processing in Autonomous Agents*. PhD. Dissertation, The University of Birmingham.

[Almeida 2002] Almeida, C. (February 23, 2002). Oscar-winning animator Chuck Jones dead at 89. *Associated Press*.

[André *et al*. 2000] André, E., Klesen, M., Gebhard, P., Allen, S., Rist, T. (2000). Exploiting Models of Personality and Emotions to Control the Behavior of Animated Interactive Agents. In Rickel, J. ed. *Proceedings of the Workshop on "Achieving Human-Like Behavior in Interactive Animated Agents,"* pp. 3-7.

[Brna *et al*. 2001] Brna, P., Cooper, B., Razmerita, L. (2001). Marching to the Wrong Distant Drum: Pedagogic Agents, Emotion and Student Modelling. In *Proceedings of the 2nd Workshop on Attitude, Personality and Emotions in User-Adapted Interaction*.

[Cañamero 1997] Cañamero, D. (1997). A Hormonal Model of Emotions for Behavior Control. Presented in the *Fourth European Conference on Artificial Life, ECAL'97*.

[Collins 1991] Collins, A. (1991). The Role of Computer Technology in Restructuring Schools. *Phi Delta Kappan*, 73:28-36.

[Doll 1993] Doll, W.E. (1993). *A Post-Modern Perspective on Curriculum*. New York: Teachers College Press.

[Elliott 1992] Elliott, C.D. (1992). *The Affective Reasoner: A Process Model of Emotions in a Multi-agent System*. PhD. Dissertation, Northwestern University, Evanston, Illinois.

[El-Nasr & Skubic 1998] El-Nasr, M.S., Skubic, M. (1998). A Fuzzy Emotional Agent for Decision-Making in a Mobile Robot. In *Proceedings of the IEEE International Conference on Fuzzy Systems.*

[FitzGerald & Lester 1997] FitzGerald, P., Lester, J. (1997). Knowledge-Based Learning Environments: A Vision for the 21st Century. In Martorella, P., ed. *Interactive Technologies and the Social Sciences: Emerging Issues and Applications.* New York: SUNY Press, pp. 111-127.

[Frijda 1986] Frijda, N.H. (1986). *The emotions*. Cambridge University Press.

[Gance 2002] Gance, S. (2002). Are Constructivism and Computer-based Learning Environments Incompatible? *The Journal of Education, Community, and Values: Interface on the Internet*. Pacific University, Forest Grove, Oregon.

[Giarratano & Riley 1998] Giarratano, J.C., Riley, G. (1998). *Expert Systems: Principles and Practices*. Boston: PWS Publishing.

[Gonçalves & Gudwin 1999] Gonçalves, R., Gudwin, R.R. (1999). Emotions: A Computational Semiotics Perspective. In *Proceedings of 1999 IEEE International Symposium on Intelligent Control, Intelligent Systems and Semiotics*.

[Gratch 2000] Gratch, J. (2000). Émile: Marshalling Passions in Training and Education. In *Proceedings of the 4th International Conference on Autonomous Agents*, pp. 325-332.

[Lange *et al*. 1981] Lange, O.L., Nobel, P.S., Osmond, C.B., Ziegler, H. eds. (1981). *Encyclopedia of Plant Physiology, New Series, Vol. 12A, Physiological Plant Ecology I.* Berlin: Springer-Verlag.

[Lange *et al*. 1982] Lange, O.L., Nobel P.S., Osmond, C.B., Ziegler, H. eds. (1982). *Encyclopedia of Plant Physiology, New Series, Vol. 12B, Physiological Plant Ecology II.* Berlin: Springer-Verlag.

[Lester *et al*. 1996] Lester, J., Stone, B., O'Leary, M., Stevenson, R. (1996). Focusing Problem Solving in Design-Centered Learning Environments. In *Proceedings of the Third International Conference on Intelligent Tutoring*, pp. 475-483.

[Lester *et al*. 1997] Lester, J., Converse, S., Kahler, S., Barlow, T., Stone, B., Bhogal, R. (1997). The Persona Effect: Affective Impact of Animated Pedagogical Agents. *In Proceedings of CHI '97*, pp. 359-366.

[Lester *et al*. 1999] Lester, J., Stone, B., Stelling, G.D. (2000). Lifelike Pedagogical Agents for Mixed-Initiative Problem Solving in Constructivist Learning Environments. *User Modeling and User-Adapted Interaction,* 9(1-2):1-44.

[Lester *et al*. 2000] Lester, J.C., Towns, S., Callaway, C., Voerman, J.L., FitzGerald, P.J. (2000). Deictic and Emotive Communication in Animated Pedagogical Agents. In *Embodied Conversational Agents*, MIT Press, pp. 123-154.

[Marsella *et al*. 2000] Marsella, S.C., Johnson, W.L., LaBore, C. (2000). Interactive Pedagogical Drama. In *Proceedings of the 4th International Conference on Autonomous Agents*, pp. 301-308.

[Martinho & Paiva 1999] Martinho, C., Paiva, A. (1999). Developing Pathematic Agents. In *Proceedings of EBAA'99: Workshop on Emotion-Based Agent Architectures*.

[Reeves & Nass 1996] Reeves, B., Nass, C. (1996). *The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places*. Stanford, California: CSLI Publications and Cambridge University Press.

[Ortony *et al*. 1988] Ortony, A., Clore, G.L., Collins, A. (1998). *The Cognitive Structure of Emotions*. Cambridge University Press.

[Osterweil 2000] Osterweil, N. (November 17, 2000). He Who Laughs, Lasts. *WebMD Medical News,* WebMD Corporation.

[Pfeifer 1988] Pfeifer, R. (1988). Artificial Intelligence Models of Emotion. In *Cognitive Perspectives on Emotion and Motivation*. Kluwer Academic Publishers.

[Piaget 1954] Piaget, J. (1954). *The Construction of Reality in the Child*. New York: Basic Books.

[Picard 1997] Picard, R. (1997). *Affective Computing*. MIT Press.

[Reilly 1996] Reilly, W.S. (1996). *Believable Social and Emotional Agents*. PhD. Dissertation, Carnegie Mellon University.

[Rickel & Johnson 1997] Rickel, J., Johnson, L. (1997). Integrating Pedagogical Capabilities in a Virtual Environment Agent. In *First International Conference on Autonomous Agents*, pp. 30-38.

[Russell & Norvig 1995] Russell, S., Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Inc.

[Sloman 2001] Sloman, A. (2001). Beyond Shallow Models of Emotion. In *Cognitive Processing*, 2(1):177-198.

[Small 1998] Small, P. (1998). *Magical A-Life Avatars*. Manning Publications Co.

[Smith & Lazarus 1990] Smith, C.A., Lazarus, R.S. (1990). Emotion and Adaptation. In Pervin, L.A. & John, O.P. ed. *Handbook of Personality: theory & research*. Guilford Press, NY, pp. 609-637.

[Thomas & Johnson 1981] Thomas, F., Johnson, O. (1981). *Disney Animation: The Illusion of Life*. New York: Abbeville Press.

[Voerman 1997] Voerman, J.L. (1997). *Deictic Behavior Control for Believable, Animated Pedagogical Agents*. Masters Thesis, North Carolina State University.

[Young 1999] Young, R.M. (1999). Cooperative Plan Identification: Constructing Concise and Effective Plan Descriptions. In *Proceedings of the National Conference of the American Association for Artificial Intelligence*, pp. 597-604.