Knowledge Systems Laboratory **December 1997**
Report No. KSL 97-09

Agents in Annotated Worlds

by

Patrick Doyle, Barbara Hayes-Roth

KNOWLEDGE SYSTEMS LABORATORY
Department of Computer Science
Stanford University
Stanford, California 94305

Abstract

Virtual worlds offer great potential as environments for education, entertainment, and collaborative work. Agents that function effectively in heterogeneous virtual spaces must have the ability to acquire new behaviors and useful semantic information from those contexts. The human-computer interaction literature discusses how to construct spaces and objects that provide "knowledge in the world" that aids human beings to perform these tasks. In this paper, we describe how to build comparable *annotated environments* containing explanations of the purpose and uses of spaces and activities that allow agents quickly to become intelligent actors in those spaces. Examples are provided from our application domain, believable agents acting as inhabitants and guides in a children's exploratory world.

Keywords: believability, human-like qualities of synthetic agents, synthetic

agents

1. Introduction

Today's virtual environments present opportunities for simulation and interaction involving many users at once, in a way impossible only a few years ago. Although virtual worlds have existed in some form for more than two decades, most have been small in size and limited in scope. Advances in communications technology and computing power are already producing rich and complex environments, and soon promise elaborate virtual worlds able to support thousands of users simultaneously engaged in many different tasks.

There are two critical technical issues in scaling such spaces from the small worlds of the past to the ambitious visions of the future. The first is incorporating considerable amounts of substantive content, and the second is providing adequate support for the maintenance and effective use of such content. In the past, human-human interaction has been sufficient to build and update these worlds as well as to meet the needs of users on a person-to-person basis. This resource will not scale to meet all of users' growing demands; instead, we will need to create intelligent agents to answer queries, provide guidance, participate in the activities of human users, and otherwise generate meaningful content through their own actions.

The abilities of such agents to learn and reason are greatly limited and currently far inferior to humans'. To offset these limitations, successful environments must themselves provide directions to guide the agents. Even human beings rely upon the real world to tell them what to do and why; well-designed objects and spaces can help us in performing daily tasks by containing clues to their meanings and operations. For agents, less sensitive and intelligent than human beings, such directions are even more critical to satisfactory performance.

We call such directions *annotations* of the environment. They are analogous to the marginalia of a textbook or the markup of a Web page, though potentially richer than either. The purpose of these annotations is to enable agents with limited understanding of an environment's context, meaning, or their appropriate behaviors within the environment to learn them through direct queries to the space rather than some reasoning process. Such annotated worlds support agents with a broad range of capabilities and goals. They also allow all agents to function effectively in harmony with the designers' intended use of the space.

In our work, we are annotating an educational children's environment for use by human-like characters with simulated emotions and personalities. Our current interest is to annotate the space so that we can insert a variety of characters with different personalities and goals that will populate the world as friends, guides, competitors and passers-by to be encountered by the children. The agents initially know nothing about the content of the world, only its basic dynamics; they rely upon the annotations for the rest.

In this paper we motivate the need for annotations in multi-user worlds and examine dialogs from our testbed to illustrate how several types of annotation are used together with a simple believable agent architecture to produce engaging, entertaining, and educational experiences for children. Section 2 examines the reasons for annotating virtual places. Section 3 outlines the potential uses of annotations in virtual worlds. Sections 4 and 5 briefly describe the types of annotations and the agent architecture that interacts with them. Section 6 discusses dialogs between a user and two believable agents, and Section 7 offers concluding thoughts.

2. Things That Make Agents Smart

- A college student walks up to an ATM. When the screen lights up, he enters his access code. When prompted, he chooses a transaction. When the deposit slot opens and flashes, he inserts an envelope. The slot closes and the transaction is complete. The student walks away.
- A woman, leading her child by the hand through a history museum, walks up to an exhibit on 18^{th} century America. Quickly scanning the printed text, she turns to her daughter and explains what a cotton gin is, and how it works.
- A man walks up to a door. There is a bar at waist height, which he attempts to push. In fact, the bar must be pulled to open the door.

Human beings rely upon the world to help them function. We do not memorize the procedure for making an ATM deposit, bring textbooks to museums to understand the exhibits, or become experts on doors in order to navigate buildings. Instead, we expect to find clues that give us the information we need as we need it. These clues enable us to act as if we had already studied and reflected upon what we encounter. Instantly, they make us limited experts in the domain. (Norman, 1990) calls this "knowledge in the world."

There are several advantages of building such knowledge in the world. First, knowledge in the world is immediately retrievable; it is right there in front of us. If the world is properly designed, we simply look (or listen, or feel) and the information we need is at hand. Second, it does not require any learning. Instead, it is only necessary to interpret the world to use it.

It is important to recognize, though, that the difficulty of interpretation depends upon how carefully that part of the world was designed. A good design facilitates immediate understanding, easing a user's cognitive and memory loads; a bad design can mislead and create precisely the kinds of problems it was intended to solve, demanding extra thought and memorization to get around it (Norman, 1993).

Third, such knowledge is always up-to-date and accurately reflects the state of the world, as memory may not. As (Brooks, 1990) observes, "The world is its own best model." Lastly, such knowledge makes it possible to use the object or environment almost immediately. No practice is required. A door handle allows any ordinary person to use it without any foreknowledge; the cockpit of an airplane, with its multitude of dials and switches, does not.

Much of the information human beings need to perform daily tasks is embedded in the world. As (Norman, 1990) writes, "People routinely capitalize on this fact. They can minimize the amount of material they must learn or the completeness, precision, accuracy, or depth of their learning. People can deliberately organize the environment to support their behavior."

There are of course disadvantages to this approach. It can lead to clumsy, cluttered designs, since much information needs to be available to the user for interpretation. In the airplane cockpit example, physical restrictions make it impossible to surround all of the controls with labels, diagrams, and detailed explanations. Relying upon knowledge in the world can also be an inefficient process. It must be found, examined, and processed before it can be used. Internal knowledge, what (Norman, 1990) calls "knowledge in the head," has already been interpreted and can be much more efficient to apply. Such knowledge makes us quick to act and precise performing well-understood tasks in familiar places.

However, the average human (or agent) today is often in complex and unfamiliar environments. It is not possible to function using only known skills and interpretations of things past. Instead, we dynamically analyze our world as we interact with it, making new interpretations of our environment and its objects and acquiring new skills. The world can either facilitate or hinder that process, both for people and for agents.

Human beings expect the world to help them act within it. One way of doing this is to design the world so that its properties make it easy to perform the intended operations, and difficult to do things that are against the designer's intention. The concept of *affordance* (Gibson, 1977**)** refers to the properties of a thing, particularly those properties that determine how a thing could be used. A chair, with its flat surface a few feet from the floor, affords sitting on, leaning on, picking up. A door handle affords pulling or pushing. A sign affords reading. A well-designed environment affords just those activities intended by the creator.

In the real world, a person's ability to perceive affordances is compromised by the noisy, uncertain nature of the environment. The designer has limited resources to convey the intended use of a thing or place, and must hope that the user correctly interprets his or her intent. A human being must overcome these obstacles in order to make use of the knowledge in the world.

An agent in a human environment is even more helpless. It does not have access to the wealth of unobtrusive but critical information that allows us to perform our tasks. The vocabulary of visual clues, physical constraints, and commonsense knowledge which the designer relies upon is not one most agents can speak.

In a virtual world, however, the designer has complete control over what kinds of knowledge is available in the world, how the agent can access it, and the form it takes. Knowledge can literally be embedded in the world as *annotations* attached to objects, entities, and locations. Such information can be more explicit and less ambiguous than in the real world. With these annotations the world-builder can provide comprehensive and exact descriptions of affordances in languages agents are well suited to understand.

3. The Possibilities of Virtual Worlds

As network connectivity becomes widespread interest is fast growing in virtual worlds as vehicles for simultaneous multi-user interaction. Such worlds fundamentally offer two things: the ability to communicate with other people in real time across great distances, and the ability to do things not possible in the real world. They promise fantastic unreal places that nonetheless seem real, in which new kinds of collaborative work, education, and entertainment are possible.

Agents must be able to explore such varied worlds just as humans can. To do this, just as in the real world, they must be able to sense and interpret the environment intelligently. But if individual agents are meant to function in a variety of virtual worlds, they may have little or no foreknowledge of the ones they visit. To function without it, they must either have sophisticated reasoning abilities or they must have help from the environment. Ideally, they will have both, but currently the latter is considerably more practical than the former.

This is the motivation behind the current research. It is our intention to design a virtual environment that will support the activities of intelligent agents by embedding abstract knowledge in the environment with which such agents can reason. This provides a base that allows agents to become instant experts on the content of the world and to retain that expertise as the world changes around them. In the future it may well be that adequately sophisticated agent systems can extract meaning from the same information presented to humans, but this is not yet a reality.

Our exploratory environments are called MUDs, or Multi-User Dimensions (or Dungeons) (Bartle 1990). We use these text-based, multi-user virtual worlds for three reasons. First, it is easy to build and annotate the environment in the MUD's internal programming language, an object-oriented variant of C[1]. Second, text is easy both to parse and to generate when compared with graphical environments; this simplifies perception and control issues, and also allows us to produce substantive and interesting worlds without elaborate graphic design. Lastly, considerable research has been done on the use of MUDs in education and believable character design, providing us valuable information on how humans interact in such worlds (Clodius 1997; Curtis 1992; Foner 1993; Kiss 1997).

4. Building Annotations

Most MUD environments contain certain basic information about objects, people, and location that can be accessed through simple queries. Thus even without additional, agent-directed annotations tailored to them, agents can extract some useful data about available actions, who and what is present at the current location, and so on. However, this simple standard data does not provide its own semantics. It is assumed either that special programs have been hard-wired to make use of the information, or that human beings with significant domain knowledge can make sense of it. Without information about meaning, explicit or otherwise, an agent cannot intelligently perform actions relevant to the domain.

The goal of our project is to provide such information in the form of detailed and agent-directed annotations. There are some basic operations that are the same in all MUDs (such as speaking), but each particular world has its own unique content. We would like the agent to learn this content in the same way as a user would: through exploration. In annotated worlds, the agent can directly query the environment about its content, available activities, recommendations for exploiting the environment given its goals and abilities, and so on. These annotations can be stored for later use, subject to any updates the environment provides as it evolves.

The large drawback, from a design perspective, of incorporating annotations in the environment is that it substantially increases the burden on the world's designers. They must provide detailed descriptions of content and action simply to support agent participation in the space. However, there are corresponding advantages to giving this task to the world builders. They know best what the salient features of the environment are and, more importantly, they understand the larger meaning of the space in a way that no current agent system could comprehend. This approach also allows them to modify the annotations as the environment changes, with the expectation that the agent will automatically adapt as it sees them. Most importantly, many agents of differing abilities and motivations can be informed by the same set of annotations.

[1] There are perhaps a half-dozen mainstream MUD systems. Our work has been done using MudOS, which provides a UNIX-like command system and object-oriented programming language based on C.

The set of annotations we are currently using is meant to support lifelike computer personalities as friends or assistants to children exploring virtual environments. There are five main types implemented to date:

- *emotional annotations*, either on environments or on events, explaining how a "typical" emotional agent might respond,
- *responsive annotations* explaining how an agent might react to events in the environment, either with domain-specific actions or suggesting types of believable behaviors built into the agent,
- *problem-solving annotations* that describe puzzles in the world, indicate hints that the agent might speak or perform, and update the agent as parts of the problem are solved,
- *role annotations* that inform the agent about actions relevant to performing certain jobs in the world so that it can combine its personality with domain-specific behaviors to function as an integrated, purposeful character, and
- *game-playing annotations* that describe the status of a game (or any bounded multi-user interaction), suggest moves an agent might make based upon its personality and desired skill level, and inform the agent how it is doing.

The first four categories are *declarative*, or static in nature; they are fixed text stored in the MUD and sent to the agent either upon demand or as events occur. For example, the agent might ask about the artist who painted a watercolor hanging in the room (*Impression: Sunrise* was painted by Monet), or it might be told about an action that takes place in its environment (Albert just kicked you). Annotations generated by actions are "fixed" in the sense that they require no real reasoning in the MUD, though they may need minor syntactic manipulations, etc.

The last category is *procedural* or dynamic, and involves substantive computation in the MUD in real-time in response to the agent's query. As an example, an agent may query the environment about its situation in a game of chess; the environment might respond with an evaluation (awful). The agent could also ask the environment for a suggested move, and given the situation the environment could offer one based upon a heuristic programmed into it (resign now). Game-playing annotations are discussed in greater detail in (Doyle and Hayes-Roth

```
EMOTION(FEAR, AVERAGE, SOURCE_ENVIRONMENT)
PROBLEM("Tomb Door Problem", "Start", HARD, 1)
HINT-TEXT("Tomb Door Problem", "Start", 1, "Opinion", "there's a clue on the map")
HINT-TEXT("Tomb Door Problem", "Start", 2, "Fact", "the Egyptians could balance enormous slabs of stone so
that they could be pivoted by a single person pushing on them")
HINT-ACT("Tomb Door Problem", "Start", 3, "push wall")
HINT-TEXT("Tomb Door Problem", "Half Open", 1, "Opinion", "you haven't pushed it in far enough")
PROBLEM-UPDATE("Tomb Door Problem", "Half Open")
EMOTION(FEAR, HIGH, SOURCE_EVENT)
PROBLEM-SOLVED("Tomb Door Problem")
```

Figure 1. Simple Declarative Annotations Embedded in the Pyramid Environment

1997).

Figure 1 shows some of the declarative annotations stored in one of our virtual spaces. They are used by the agent in the dialog of Figure 2. This brief set of extremely simple annotations allows the agent to integrate with the environment to generate purposeful, domain-dependent behaviors.

5. A Simple Believable Agent

We have constructed a simple believable agent architecture to explore our annotated MUD. There has been much interest recently in creating believable agents, including study of emotional models (Bates 1994; Elliott 1997; Ortony, Clore & Collins 1988), improvisational storytelling (Hayes-Roth et al. 1994), modeling living entities for full-body human interaction (Maes 1995), and pedagogy (Stone and Lester 1996).

Our system uses a simple emotional model based upon that in (Hayes-Roth et al. 1994). Agents have a fixed set of integer-valued emotions (such as *happiness* or *fear*) which are influenced by their own actions and by perceptions of the environment and the actions of others. Emotional personality is formed by a set of emotions together with normal values toward which they tend over time.

The actions available to an agent fall into two categories. The first is what (Lester and Stone 1997) calls *believability-enhancing* behaviors: those that have no substantive effect on the environment or the situation but stimulate the user's belief in the sentience and intention of the character. These behaviors are built into the agent and function independently of the environment. Examples include emotional expressions (smiling, frowning, kicking another agent), commentary (on the agent's mood, the weather, topics of interest to the agent), or inwardly-directed actions (searching for a pair of glasses).

The second category is the *domain behaviors*. These are explicit domain actions or suggestions for appropriate believability-enhancing behaviors that are given to the agent through annotations. The agent contains no such behaviors *a priori*, but gains them through exploration of the virtual world. Examples might be experimenting with devices in a scientists laboratory, playing a game of chess, or lecturing on the history of Egyptian hieroglyphics.

The set of domain behaviors currently available depends on the agent's location and on the state of the environment. Annotations may either add or remove behaviors from an agent's repertoire. So, for example, if an agent enters a room with an open door, it has the option of closing it; but, once closed, it loses that action and instead gains an action for reopening it.

The agent's basic control cycle consists of the traditional sense-think-act loop. During sensing the agent confirms its current location in the world. If its location has changed, it removes location-dependent annotations to a permanent database. It tests the current environment to see whether it has been there before, and if so whether its annotations have been updated since its last visit. If the space is unchanged, it retrieves its annotations from the permanent memory; otherwise, it queries the environment and makes a new entry for these annotations. It also processes any annotations it has received since the last sensing operation and attends to them so, for example, if a user has kicked the agent, it modifies its emotions appropriately (it gets angry).

The thinking cycle determines which action to execute using a competition-based system similar to that of (Lester and Stone 1997). Each possible legal action, both believability-enhancing and domain-dependent, has a certain amount of energy. The action with the highest energy is chosen for execution during each round. One source of energy is the appropriateness of each action with respect to current emotion values. Another is a fixed amount of energy given to each action category, representing a preferred distribution across types of actions so, for example, in the absence of any emotional influence problem-solving behaviors receive more energy than believability behaviors. When a behavior is chosen, the energy levels of all behaviors in its category are reset. This straightforward process allows us to integrate preprogrammed, internal behaviors with learned, external behaviors in a way that produces reasonable behaviors in practice.

6. Sample Interactions

Figures 2 and 3 show interactions between our simple agent and a user in an annotated MUD

```
> look
The Antechamber
   This is a dark and musty room, feebly lit by a flickering torch.  According to the ancient
map, this is the antechamber of the tomb, and somewhere on the other side of the south wall is
the entrance to the burial chamber itself.  The walls are covered with once-brilliant images of
the Egyptian gods and passages from that sacred text, the Book of the Dead.
   The only visible exit is the passage to the west.b

--  There is one obvious exit:  west.

Harker fidgets.
> scratch
You scratch your head in puzzlement.
Harker says: This might not be such a good idea.
> examine walls
The walls are covered with carvings and paintings depicting the Pharaoh’s passage to the
Underworld.
> think
You think carefully.
Harker says: Maybe there’s a clue on the map.
> ask harker for map
Ok.
Harker gives you an ancient map.
> examine map
This is an ancient map of the tomb of Pharaoh Menechepere.  It shows the antechamber, which is
adjacent to the burial chamber itself.  It does not show a connecting door, but drawn next to the
antechamber is an udjat, otherwise called the Eye of Horus.
Harker clears his throat nervously.
> examine south wall
The south wall depicts the ritual of the Opening of the Mouth.  Among the many hieroglyphs
surrounding the image of the mummified Pharaoh is the Eye of Horus.
Harker thinks carefully.
Harker asks: Come on, boss, let’s go, OK?
> give map to harker
Ok.
Harker says: I’ll keep an eye on this.
Harker says: It’s rumored that the Egyptians could balance enormous slabs of stone so that they
could be pivoted by a single person pushing on them.
> push eye
You press on the Eye.  It squeaks and sinks slightly into the wall.
Harker clears his throat nervously.
Harker says: Maybe you haven’t pushed it in far enough.
> push eye
You push again on the Eye.  With a resounding grating sound, a section of the south wall pivots,
revealing the dark entrance to a room beyond!
Harker asks: Are you sure this is such a good idea, boss?
Harker shivers nervously.
```

Figure 2. Dialog with Harker inside the Pyramid

environment. In Figure 2, the user takes the role of an intrepid explorer searching for a secret entrance into the heart of the lost pyramid of a fictional Egyptian Pharaoh. The agent, called Harker, plays the user's assistant on this expedition.

The Harker character is given a timorous personality (by default his level of fear is high), so he is inclined to make fretful remarks. This tendency is exacerbated by the environment – deep within an ancient pyramid – to make him more than usually scared. As a result, his worried comments dominate his other actions, and the cheerful side of his personality exhibited by friendly actions is suppressed.

Harker is doing three things in the course of this interaction. The first, and simplest, is responding to user commands. When asked for the map, he hands it to the user, and when the user hands it back to him he takes it. This is accomplished through PERCEIVED-ACTION annotations that inform him taking/giving operations are occurring, and Harker has a standard repertoire of responses for such situations.

The second thing Harker is doing is performing believability-enhancing behaviors, such as fidgeting or begging to leave the room. These behaviors are part of Harker's basic actions, and do not come from the room; however, they are enhanced by an EMOTION annotation on this room indicating that it generates higher-than-usual fear. He responds particularly fearfully when the user opens the secret door ("*Are you sure this is such a good idea, boss?*") because that action generates an event-caused EMOTION of high fear. Note that these annotations do not force Harker to respond fearfully; if he had an especially brave personality, he might laugh in the face of danger. The point is that they indicate how a typical agent would react.

Harker's third task is to provide hints about the puzzle of opening the secret door. As can be seen from the transcript, the solution is to press the Eye of Horus until it slides into the wall, and the diagram on the ancient map contains a clue to this solution. However, in case the user needs help, the room is annotated to enable Harker to give it. A PROBLEM annotation describes the puzzle, and HINT-TEXT and HINT-ACT annotations suggest things the agent might say or do to give clues to the user. HINT-TEXT annotations are clues meant to be spoken by the agent, and are annotated either as FACTs (e.g., heavy slabs of stone can pivot easily) or OPINIONs (there might be a clue on the map). Simple natural language processing is performed on the verbal clues to give them the agent's idiosyncratic flavor of speech.

Note that in this interaction the agent gives up its hints quite quickly. The speed with which it does this can be altered based upon the difficulty of the problem and the user's level of sophistication. (In this case it was set high to produce a brief dialog.) If the problem were a complex one that involved many locations and/or states, the agent might absorb hints from many places and dole them out more sparingly. As it is there are only three states in this puzzle: the initial state, the state where the Eye has been pressed once, and the solution state. When the Eye is pressed the first time, a PROBLEM-UPDATE annotation is sent, causing the agent to change its choice of hints.

In the second transcript, the user is talking to an agent for the more prosaic purpose of buying bread. The agent, Antonio, has been told to play the role of baker. This means it will be sensitive to any ROLE annotations in the environment giving specific information about how to act as a baker would. Antonio has a warm, friendly personality, and in the absence of any emotional annotations in the room he exhibits generally happy behaviors.

```
> look
The Bakery
   This large, bright room is permeated with the delicious smells of breads, cakes, pies and
other comestibles.  A long, low counter divides the room in half; on the far side are the ovens
and worktables where the food is made.  Sitting on the counter are boxes and cases displaying the
goods for sale.
   Currently there is nothing for sale.

--  There is one obvious exit:  west.

> say Good morning!
You exclaim: Good morning!
Antonio exclaims: Good morning!
Antonio turns the oven on.
Antonio casts an eye around the kitchen.
Antonio looks out the window.
Antonio says: Time to make the doughnuts.
> buy bread
You ask Antonio for some bread.
Antonio says: We haven't got any bread yet, but should have some fresh soon, if you'd care to
wait.
Antonio grins mischievously.
> nod
You nod solemnly.
> Antonio says: It's a lovely day outside.
Antonio puts some flour in a bowl.
Antonio hums softly to himself as he works.
Antonio smiles at you.
Antonio cracks some eggs into the bowl.
Antonio adds some water to the bowl.
Antonio whistles merrily.
Antonio mixes the ingredients.
Antonio wipes his brow and says, 'This is hard work!'
Antonio puts the dough on the worktable.
Antonio grins mischievously.
Antonio kneads the dough into loaves.
Antonio puts the loaves into the oven.
Antonio waits a while.
Antonio casts an eye around the kitchen.
Antonio takes the loaves out of the oven and puts them on the counter.
> buy bread
You ask Antonio for some bread.
Antonio hands you a loaf of bread, saying 'I hope you enjoy it!'
```

Figure 3. Dialog with Antonio in the Bakery

Antonio is doing two things in this text. The first is his reaction to the user's comments and queries; he response to a "Good morning!" with a "Good morning!" of his own, and when the user attempts to buy bread, he can either say that it's not ready yet or sell some. His response depends upon the state of the environment; when the bread has been baked, the environment sends out an ADD-RESPONSE update that changes his response to "buy bread" from apologizing to selling. Similarly, when the bread runs out it is changed back again.

Antonio is also executing two *threads* (or *trains of thought*) simultaneously. The first and longer one is the bread-making process, made up of actions such as turning the oven on, putting flour in a bowl, and so on through putting the bread on the counter. The other, scarcely noticeable thread consists of looking out the window and commenting on what a lovely day it is. The window thread is built into Antonio's personality, while the baking thread is provided by a THREAD annotation in the room, coupled with the ROLE annotation for the baker. The THREAD annotation suggests a sequence of actions following a common theme; they do not have to be performed contiguously but they will be performed in sequence.

Interspersed among the actions of each thread are his friendly, atomic believability-enhancing behaviors. A level of "binding energy" is associated with each thread to indicate how important it is to keep executing its actions in sequence without interleaving other acts; in this case neither thread has a high level (unlike a thread for open-heart surgery might be, for example). Threads (such as looking for one's glasses or a giving running commentary on the weather) give a sense that the agent is thinking about a single topic over a span of time, rather than executing a series of unrelated actions which will quickly dispel the illusion of life.

7. Conclusion

To successfully use agent systems in virtual worlds, we must have well-designed agents and well-designed environments. Each is a complex and demanding task. By building "knowledge in the world" for our virtual spaces, we can separate domain knowledge from the agent architecture and provide an interface through which work on each problem can be made to communicate. The challenge is to evolve the agents and virtual worlds together so that the annotations fuel the natural capabilities of the agents, just as well designed objects in the real world afford intelligent use by people.

In this paper we have outlined the beginning of one such evolution. We have described a simple believable agent architecture, and constructed a virtual world containing annotations that supports its believable behaviors and enhances them with content-sensitive actions. The result is a set of simple agents that produce entertaining, educational, and believable behaviors.

Acknowledgments

Our work has been supported by NSF Contract IRI-9424261, ARPA Contract N66001-95-D-8642-Subcontract #137-1 through Teknowledge, Inc., gifts from Intel, and Seed Grants from Stanford's Center for the Study of Language and Information and Office of Technology Licensing. We thank David Schairer, administrator of Overdrive MUD, for providing a testing environment, and Lee Brownston and Daniel Rousseau for illuminating discussions on our designs.

References

Bartle, R. 1990. Interactive multi-user computer games. MUSE Ltd. Research Report.

Bates, J. 1994. The Role of Emotion in Believable Agents. Communications of the ACM 37, no. 7 (July): 122-125.

Bates, J., Loyall, A.B., and Reilly, W.S. 1992. An Architecture for Action, Emotion, and Social Behavior. Technical Report CMU-CS-92-144. Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA. (May).

Brooks, R. 1990. Elephants Don't Play Chess. Robotics and Autonomous Systems, no. 6: 3-15.

Clodius, J. 1997. Creating a community of interest: "Self" and "Other" in DragonMUD. Combined Winter Conference on Educational Uses of MUDs (Jackson, WY, Jan. 12-17).

Curtis, P. 1992. Mudding: Social Phenomena in Text-Based Virtual Realities. In Proceedings of the 1992 Conference on Directions and Implications of Advanced Computing. Berkeley, CA.

Doyle, P. and Hayes-Roth, B. 1997. Guided Exploration of Virtual Worlds. Technical Report KSL-97-04, Knowledge Systems Laboratory, Stanford University, Stanford, CA. (May).

Foner, L. 1993. What's An Agent, Anyway?: A Sociological Case Study. Agent Memo 93-01, Agents Group, MIT Media Laboratory, Massachusetts Institute of Technology, Cambridge, MA. (May).

Gibson, J. 1977. The Theory of Affordances. In *Perceiving, Acting, and Knowing: Toward an Ecological Psychology*, R. Shaw and J. Bransford, eds. Hillsdale, NJ: Lawrence Erlbaum Associates.

Hayes-Roth, B., Sincoff, E., Brownston, L., Huard, R., and Lent, B. 1994. Directed Improvisation. Technical Report KSL-94-61, Knowledge Systems Laboratory, Stanford University, Stanford, CA. (Sept.).

Kiss, P. 1997. Evaluation of MUDs and the role of intelligent agents. Combined Winter Conference on Educational Uses of MUDs (Jackson, WY, Jan. 12-17).

Lester, J., and Stone, B. 1997. Increasing Believability in Animated Pedagogical Agents. In Proceedings of the First International Conference on Autonomous Agents (Marina del Rey, CA., Feb. 5-8). ACM, New York, NY. (Feb.)

Maes, P. 1995. Artificial Life Meets Entertainment: Lifelike Autonomous Agents. Communications of the ACM 38, no. 11 (Nov.): 108-114.

Norman, D. 1990. *The Design of Everyday Things*. New York: Doubleday.

Norman, D. 1993. *Things That Make Us Smart*. Reading, MA: Addison-Wesley.

Ortony, A., Clore, G., and Collins, A. 1988. *The Cognitive Structure of Emotions*. Cambridge University Press, Cambridge, UK.

Stone, B., and Lester, J. 1996. Dynamically Sequencing an Animated Pedagogical Agent. In Proceedings of the Thirteenth National Conference on Artificial Intelligence (Portland, OR, Aug. 4-8). AAAI, Menlo Park, CA., 424-431.