Virtual Intelligence from Artificial Reality: Building Stupid Agents in Smart Environments

Patrick Doyle
Knowledge Systems Laboratory
Gates Computer Science Building 2A
Stanford University
Stanford, CA 94305-9020
pdoyle@cs.stanford.edu

Abstract

As the nature of online gaming changes to accomodate vast, persistent virtual worlds, there is a growing need to populate them with inhabitants capable of dynamic, adaptable, *fruitful* interactions with players. In this paper we argue that by separately placing abstract compentencies in the agent and concrete domain knowledge in the environment, we can build simple intelligences that exhibit sophisticated behaviors.

Introduction

The technological advances of the last decade in networking and processor power have brought a growing number of computer games online. The playing public now expects new combat or simulation games to have multiplayer modes with at least a few participants. We are also beginning to see massively multiplayer games such as *Everquest*, *Asheron's Call*, and *Ultima Online*, that support hundreds or thousands of users simultaneously.

The advantage of these massive world is that the players themselves are the source of its constant change and growth. The environment does not need to be altered often or in significant ways to keep them interested. Play can vary from physical activities such as sailing the oceans, climbing mountains and exploring the dungeons' depths to the social experiences of building careers, engaging in local politics, and creating cities, guilds and laws through collaborative effort. If the game lacks a clear direction and a definite end, the other players keep it from becoming dull or unfocused.

The potentially serious disadvantage is that players, for all their sociability and dynamism, are rarely as well-suited to supporting the game's atmosphere and goals as the designers' carefully crafted inhabitants. In a single-player game, there is no one to break the player's immersion in the story; in a multiplayer game, almost anyone can, and probably will. Almost any player, that is.

Intelligent Agents of Change

We are researching how to improve the believability of intelligent agents, particularly agents embodied in online worlds. Currently many such creatures have severely limited reasoning abilities, no capacity to learn, and no comprehension of their context. The result is that they are either moving targets who are mowed down or they are a moving background that is largely ignored. They rarely act in ways users would think of as intelligent.

Non-player characters (NPCs) are most obviously in need of improvement. They are generally a stable population of agents that look and act like other players. They often perform mundane tasks, such as running the shops and training the players. Their main value comes from providing a sense of social continuity; they are the neighbors, laborers, sidekicks and evil geniuses that make up the society in which the players' characters live.

We believe that good NPCs are essential in immersive online games for three reasons. First, although ideally the game world would be filled with players willing to take on all the roles the society needs from street-sweeper to senator, In reality most players create characters on the fringers of society and prefer practicing knife-throwing to double-entry bookkeeping. Someone has to take on the supporting roles, and that's what NPCs are for.

Equally importantly, many gamers are not adept at role-playing, and even if they would like to behave in character they don't know how to go about it. This isn't a problem in single-player games, but in our experience, multiplayer worlds are awash with users who have all the right trappings – the equipment, the abilities, the sounds and graphics – but aren't able to foster the experience they're looking for. We play games because they tell compelling stories. Stories depend on emotion (Elliott *et al.* 1998), and emotion depends upon character. Intelligent NPCs are well-designed characters that can provide atmosphere as well as examples to emulate. They draw both the player and the character into the story.

Finally, NPCs are the active representatives of the designers. They direct players to the unusual, interesting, engaging parts of the environment. The physical layout of the game world permits characters to perform certain actions; the social layout provided by the NPCs motivates those actions, in the form of quests, legends, economics, and so on. NPCs are an important way in which the world makes itself interesting to player characters.

Making traditional NPCs more believable can be done in two ways. The first is to improve their believability-enhancing behaviors, those that don't directly influence the world but give us more insight into their personalities and moods. Considerable research has already been done in this area (see (Hayes-Roth & Doyle 1998) for a survey). The second is to make the characters more intelligent about the world, by improving what we call their domain behaviors. When they act reasonably in the changing environment, players will be more willing to ascribe inner motivations to their actions. The belief in these inner motivations makes these characters compelling.

This is the goal of our current research. We are building virtual worlds that will support the intelligent activities of limited agents. Our approach is to embed abstract knowledge in the environment that these agents can extract and use for guidance. This allows these agents to become instant experts on the content of the world no matter where they travel, and to retain that expertise as the world changes around them.

What Designers Can Afford

Much of what people need to know to perform routine tasks is embedded in the world. As Norman writes (Norman 1990), "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."

In fact, people expect the world to help them act within it. The artifacts we use are designed so that their properties make it easy to use them as the designer intended, and difficult to do things that weren't intended. 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 – a flat surface a few feet from the floor – affords sitting on, standing on, leaning on, picking up. A door handle affords turning. A button affords pushing. A well-designed environment affords just those operations it is meant to support.

In the noisy, uncertain real world, affordances may be difficult to perceive. In a virtual world, however, the designer has control over what knowledge is available and how it is accessed. Affordances are literally embedded in objects, entities, and locations, and can be more detailed and less ambiguous than in the real world.

We call these virtual affordances *annotations* of the environment. As an example, they are analogous to, though more powerful than, the markup of a Web page. The purpose of these annotations is to enable agents with limited knowledge of an environment to learn about it through direct queries rather than pure reasoning. Annotations can support agents with a broad range of goals and abilities to act in harmony with the intended use of the space.

This approach has several advantages. First, knowledge about the world is always up-to-date and accurate; agents immediately adapt to changes in the world. As Brooks observes, "The world is its own best model." (Brooks 1990) Secondly, it is immediately available and only need be interpreted to be used. Lastly, no practice is required to use the knowledge. A door handle allows any ordinary person to use it without training; an airplane cockpit, with all of its dials, levers, and procedures, does not. In an annotated world, even a very limited agent could use both immediately.

Of course, there are disadvantages as well. Annotations can lead to large, complex representations of the environment, since a great deal of information must be available to the agent. It must be transferred and processed before it can be used, which is slower and less efficient. It is useless to an agent that cannot comprehend it. Yet we believe this is a valid approach for complex online worlds with many agents whose competence directly influences the value of the users' experiences.

Research

In our work, we have been annotating a virtual environment populated by believable characters with simulated emotions and personalities. These characters act as friends, guides, competitors, and inhabitants that the users will encounter. Our goal is to annotate the world to support many characters with different personalities, goals, and basic competencies, but which are all able to use the annotations effectively. Initially, the characters only know the world's basic dynamics. The annotations provide everything else.

Our testbed is a text-based, multi-user platform called a MUD (Multi-User Dungeon or Dimension) (Curtis 1992), specifically, MudOS. The MUD's internal programming language, an object-oriented variant of C, makes it easy to build and annotate the environment. Also, text is easy to generate and to parse, which simplifies perception and control issues, and allows us to build interesting worlds without elaborate graphics.

Currently we have five types of annotations for believable characters:

- *emotional annotations* on environments or on events, indicating 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 problems 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 structured 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; they are text stored in the MUD and sent to the agent either on request or as the world changes. An agent placed in a laboratory and told to take on the role of a mad scientist would ask for role annotations that explain appropriate behaviors (cackle frequently, drooling is recommended), procedures (knock the player on the head, strap him to the table, pull the switch) or might receive responsive annotations as the player reacts (Igor just kicked you).

The last category is procedural (or dynamic), and involves real-time computation in the environment. For example, a chess-playing agent may ask the chessboard how the game is going, and the game could respond with an evaluation (awful). The agent could then ask for a uggested move given the skill level it emulates, and the environment could offer one (resign now). A preliminary discussion together with an example trace of game-playing annotations is in (Doyle & Hayes-Roth 1997).

Figure 1 shows part of a trace of an interaction between a player and an agent in one of our environments, a lost Egyptian pyramid. The player's goal is to find the burial chamber, which is hidden behind a secret door in the antechamber. The agent plays the role of Harker, a faithful if fearful sidekick, who carries the equipment and offers insight and hints during play. To keep the trace short, Harker was told to offer hints frequently.

When Harker entered the room, he knew nothing about the puzzle or its solution, and he was exhibiting no fear. He did, however, have a framework for representing state-based problems in the environment, and he was capable of processing emotional annotations. The antechamber indicated that the room itself was unsettling for a "typical" emotional agent and that certain events, such as the opening of the door, were especially frightening. Since Harker is by default more edgy than most, the result of absorbing these annotations was that he became increasingly anxious and exhibited nervous behaviors often. A bolder character might eagerly rush in where Harker fears to tread, while a robotic assistant would be unaffected by the atmosphere at all.

The room also provided all the details about the puzzle, how it could be solved, and which hints would be useful at what points. The core of these hints is embedded as fragmentary English text in the annotations, which the agent coats with syntactic sugar before uttering them. As the state of the puzzle changed, more annotations were broadcast with new hints, emotions, and appropriate actions.

If the player had made no progress, the annotations made Harker capable of solving the puzzle of the door on his own, with appropriate comments as he did. Notice that it is immaterial whether the problem is opening a secret door, launching a spaceship from Mars, or even putting all twenty treasures in the trophy case. Harker could do them all – and so could any other agent with his abilities, but each in a way in keeping with its distinct personality and behaviors.

For a more detailed analysis of Harker, the annotations in the pyramid, and this and another dialog, see (Doyle & Hayes-Roth 1998).

Applications

The applicability of this approach to computer games is obvious (at least from within the armchair of academia). Online worlds are already growing to enormous size, but the sense of adventure they promise is being overwhelmed by the large number of players who are unable to match their actions or dialog to the roles they purport to play. The result is an environment in which the physical space and the social space are distantly related at best.

By embedding annotations in the environment, we enable NPCs to travel beyond the narrow confines of their preprogrammed behaviors. They can acquire new knowledge and abilities wherever they travel, and they can continue to provide novel and meaningful interactions with the users they encounter. This makes the game more compelling in spite of other players. It also provides a social framework in which it is easier for other players to get in character and stay there, making it more compelling because of them.

Our argument is based upon the quality of the player's experience. There are other advantages. Annotations allow the character's personality to be decoupled from its knowledge, which can simplify design. If characters retain some of the annotations they perceive, they naturally evolve as unique individuals with experiences, personalities and information distinct from other characters. No two butchers, bakers, or candlestick-makers have to be alike either in what they do or how they do it. (A baker who's just seen the entire village slaughtered by demons might be permanently changed by the experience.) Finally, characters do not need reprogramming if the environment alters in little ways; they will automatically adjust to any changes as they encounter them.

The idea of annotating the environment could as easily be applied to a single-player boxed game, of course, but there are several reasons it is particularly applicable to large, persistent worlds. Because the content is server-based, the environment will change over time, and updating the annotations is easier than reprogramming the agents. Since these worlds are so large, it is unrealistic to embed detailed knowledge about the

```
> 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.

--  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 1: Dialog with Harker in the Pyramid

whole place in every agent; having them learn about just the areas they encounter is more practical and has the added benefit of naturally creating a population of NPCs with different experiences. This is also more a more natural mechanism for dealing with the deaths of old NPCs and the creation of new ones. Lastly, players will tend to have many brief interactions with NPCs, so high believability over the short term, which even brief annotations can provide, is more important than moderate believability over the long term.

Issues

The short trace in Figure 1 suggests how a small number of annotations, combined with an agent of limited intelligence, can produce behavior that is relevant and entertaining. It also demonstrates the areas that need more work:

- Limitations of "canned" annotations. Providing an agent with pre-scripted text to recite or a fixed list of actions to perform severely limits its durability, and to some extent compromises its uniqueness. This is a limitation of our current system when it provides hints and contextual commentary. However, it is not a limitation in principle; ideally, annotations would provide only the deep semantic data rather than a surface representation, and allow the agent to translate it into utterances or actions.
- The knowledge representation issues involved with widespread annotations. It is important to recognize that our goal is not a general knowledge representation language for relating action, time, entities, etc. Our thesis is that, within specific domains and for particular goals (such as allowing a believable agent to explore an educational environment), we are already capable of creating representations that will enable agents of limited intelligence to support those goals in effective ways.
- Integrating believability-enhancing behaviors and domain behaviors. The fundamental problem for our approach is building a mechanism that integrates an agent's built-in behaviors with annotation-provided behaviors in a seamless and useful way. Currently, some annotations (such as the emotional annotations) give broad suggestions about how to choose among built-in behaviors. Also, by categorizing all of an agent's behaviors (i.e., believability-enhancing, problem-solving, role-playing, etc.) we are giving it some ability to reason about how they interact whether or not they are built-in.

Conclusion

Massively multiplayer online games have great potential for growth, but only if they can meet the high expectations of their players for immersivity and constant change. By annotating these environments, designers provide affordances that allow NPCs of many levels of sophistication and of diverse abilities to behave relevantly, usefully and entertainingly. Such characters also provide a much-needed social component to reinforce the player behaviors that make the game an enjoyable place to be.

Acknowledgments

This work was supported by NSF Contract IRI-9424261, ARPA Contract N66001-95-D-8642-Subcontract #137-1 through Teknowledge, Inc., gifts from Intel, an Intel Foundation Fellowship, and Seed Grants from Stanford's Center for the Study of Language and Information and Office of Technology Licensing.

References

Brooks, R. 1990. Elephants don't play chess. *Robotics and Autonomous Systems* 6:3–15.

Curtis, P. 1992. Mudding: Social phenomena in text-based virtual realities. Available from ftp://ftp.lambda.moo.mud.org/pub/MOO/papers.

Doyle, P., and Hayes-Roth, B. 1997. Guided exploration of virtual worlds. Technical Report KSL-97-04, Knowledge Systems Laboratory, Stanford University, Stanford, CA.

Doyle, P., and Hayes-Roth, B. 1998. Agents in annotated worlds. In *Proceedings of the Second International Conference on Autonomous Agents*, 173–180. Minneapolis, MN: ACM Press.

Elliott, C.; Brzezinski, J.; Sheth, S.; and Salvatoriello, R. 1998. Story-morphing in the affective reasoning paradigm: Generating stories automatically for use with "emotionally intelligent" multimedia agents. In *Proceedings of the Second International Conference on Autonomous Agents.* Minneapolis, MN: ACM Press.

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

Hayes-Roth, B., and Doyle, P. 1998. Animate characters. *Autonomous Agents and Multi-Agent Systems* 1:195–230.

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