VR/IS Lab Virtual Actor Research Overview

Dan M. Shawver
Sharon Stansfield
Sandia National Laboratories
Albuquerque, NM 87185-0951
(505) 844-6384 FAX (505) 844-6161
shawver@isrc.sandia.gov
sastans@sandia.gov

June 22, 1995

1 Introduction

This overview presents current research at Sandia National Laboratories in the Virtual Reality and Intelligent Simulation Lab. Into an existing distributed VR environment which we have been developing, and which provides shared immersion for multiple users, we are adding virtual actor support. The virtual actor support we are adding to this environment is intended to provide semi-autonomous actors, with oversight and high-level guiding control by a director/user, and to allow the overall action to be driven by a scenario. We present an overview of the environment into which our virtual actors will be added in Section 3, and discuss the direction of the Virtual Actor research itself in Section 4. We will briefly review related work in Section 2. First however we need to place the research in the context of what motivates it.

The motivation for our construction of this environment, and the line of research associated with it, is based on a long-term program of providing support, through simulation, for *situational training*, by which we mean a type of training in which students learn to handle multiple situations or scenarios. In these situations, the student may encounter events ranging from the routine occurance to the rare emergency. Indeed, the appeal of such training systems is that they could allow the student to experience and develop effective responses for situations they would otherwise have no opportunity to practice, until they happened to encounter an actual occurance. Examples of the type of students for this kind of training would be security forces or emergency response forces. An example of the type of training scenario we would like to support is given in Section 4.2.

The requirements of situational training include multiple participants and intelligent simulation. In addition, we are concentrating our attention on "close quarters" situations, involving individual and team level action, and requiring participants to be able to interact with each other as perceivable human figures in the training setting, with "fine grain" body movements. The system we have developed already allows multiple participants to appear in embodied form within a common, shared virtual environment.

2 Related Work

2.1 Training and Simulation Systems

Currently, training using simulation (as in military or flight training) uses predominately vehicles controlled by the trainees, and perhaps by trainers, with some research work on semi-autonomous vehicle-based forces. Recent work involving automated or semi-automated vehicle-based forces includes the work of Zyda [29], Pfefferman [21], the thesis by Mohn [14], and the recent paper by Tambe et al [24].

2.2 Agents Research and Situated Agents

One of the most relevant areas of existing work is that which has been done with *situated agents* [12, 7, 8], much of which is in the area of robotics. Brooks [7, 8], for example, has proposed a radical re-thinking of robotics research that stresses both individual agents, using the *subsumption architecture*, and their situatedness as the basis for intelligent behavior.

However, this body of work deals with a more general problem area than we face. There, the desire is to eventually achieve a measure of autonomy for robot devices that will be effective in real environments. In addition, there is a relationship between this work and computer simulation work on agents, where the goal is to clarify the nature of intelligence and further the long-

This work was supported by the United States Department of Energy under Contract DE-AC04-94AL85000.

DISCLAIMER

DISCLAIMER

term goals of AI, as in the recent work of Nilsson [16, 15], Agre [1, 2], and others in the AI community.

Perhaps there is also a relationship between our interests and the field of *artificial life*. However, for our research, the simulation is a means of presentation to human users; it need not be self-sustaining, like a living ecology, nor does it have to restrict itself to techniques that extend to real robotic environments. We hope to take advantage of these limitations to create *realistic-appearing* humans; we can not hope to truly emulate human capabilities. Systems that exhibit capabilities that are closest to our goals are those systems built to present users with an interactive (but not necessarily realistic) world, such as Maes' ALIVE [13] and the Oz project [4, 11]. The former of these uses mechanisms for situated agents that will be discussed below in Section 2.2.1. The latter will be discussed in Section 2.2.4.

2.2.1 Action Selection Mechanisms and Reactive Planning

One can meet the requirement for reactive planning in a system providing simulation capabilities by using a variant of the approach first developed by Maes [12] and developed into a distributed form in the work of Zeltzer and Johnson [27, 28]. Briefly, in this approach, a virtual actor is a network of skill agents [1], each of which is defined with a set of preconditions and a set of postconditions, which are used to implicitly define a network of the agents, where predecessors of an agent are those agents that affect the preconditions of the agent, successors those that are affected by the postconditions of the agent. The preconditions are specified by a *condition list*, the postconditions are specified by a pair of lists, the *add list* and the *delete list*. This set of lists reflects that of the STRIPS planner. There are also agents (or network nodes) for sensors and for goals. In addition to the predecessor and successor links, there are *conflictor* links, for providing mutual exclusion where needed. Zeltzer's version of the skill network also allows *follower* links, to provide a means of greater focus in the activity of the net.

Simulation procedes by allowing activation energy to flow through the graph, and this energy affects the potential of an agent to become active. The agent can not, however, actually become active unless both its preconditions are satisfied and it has reached a threshold of activation energy. The links determine the paths along which energy can flow; the flow can be facilitating or inhibiting, depending on the type of link. The effect is called *spreading activation*. The flow of activation energy is such that both forward-chaining and backward-chaining effects occur. The net allows incremental planning to take place, but there is no global planning.

[1] The terminology varies with the cited papers; we use some of Zeltzer's terminology.

The sensor nodes represent sensory input to the actor; the goal nodes allow a means of specifying what should be accomplished. The world model is available to the actor only through the sensor nodes, the world model may in turn be changed by actions taken when agents are activated. The goals are set by means outside the actor-world system; for example by the user or programmer of the actor system.

Maes's networks have recently been criticized by Tyrrell [26], with the claim that they do not deal well with certain selection problems that model real-world animal selection activities. Tyrrell discusses an alternative mechanism, also network-based, which is better at the activities he has studied.

Alternative approaches for simulation of reactive, situated actors have also been developed by Bates and Loyall [11], Becket and Badler [5], the Thalmanns and their group [17], and Booth et al [6]. The first of these does not do any actual planning, although it does allow a range of actions to be reactively invoked, and supports the implementation of simulated simple actors that have an extensive repertoire of behaviors and include simulated emotional states. The system appears to make programing action sequences, as behavior segments, relatively straightforward. (This is not at all the case for the skill agent network approach.) The second system uses a network of elements (PaT Nets) to get reactivity. There is a higher-level, nonreactive planning component. The Thalmanns have explored some behavioral features in conjunction with non-real-time synthetic actors, and they use a reactive selection of (fine-grain) strategies in association with synthetic vision in the cited work.

The work of Booth et al proposes a design for a *state machine engine*, which hierarchically combines state machines and constraint resolution mechanisms. This mechanism is described more fully in Ahmad et al [3].

In general, systems such as those developed by Zeltzer and Johnson, Bates and Loyall, and Becket and Badler assume an underlying stratum that deals with continuous, feedback-controlled domains, and provides a set of constituent actions (perhaps constituted from smaller primitive actions), which set of constituent actions are invoked by the reactive planning component. That is, they separate the creation of single, continuous actions from the selection and invocation of those actions. The teleo-reactive programs of Nilsson [16, 15] allow dealing with both of these aspects of action in one formalism, and provide multiple levels of more detailed specification through procedural abstraction.

2.2.2 Individual Behaviors and Expressive Movement

Recent work by Perlin [19, 20] has shown that to an interesting extent, relatively simple kinematic techniques can create movement that is both natural and expressive, the latter being made apparent through the example of a dancer figure animated by his techniques. As described, this approach implements only a part of what we would need, but we view it as a possible underlying layer to other techniques reviewed here, and are currently experimenting with it (see Section 4.3.

2.2.3 Group Behaviors, Collision Avoidance and Navigation

The earliest work related to the issues of inter-individual interaction appears in Reynold's work on flocking [22], which provides a model for creatures forming a simple, homogeneous society, such as a (simplified) bird flock. His techniques allow reactions to other individuals to propagate through the group of simulated actors.

Recent work has looked at more realistic fish group behaviors. Fuzzy Cognitive Maps are used by Dickerson and Kosko [9] to control interactions between simulated dolphins, fish, and sharks in a simple 2D world. Full 3D worlds with simulation of physics of swimming and multi-species behaviors is included in the work of Tu and Terzopoulos [25]. The simulation itself, in the latter case, can be real-time for a moderate number of individuals.

In groups such as flocks and schools, collision avoidance for other individuals is part of the basic behavior set that allows the flock or school to form. Collision avoidance between individuals and the static parts of the environment could be handled with *gradient maps*, which have been proposed for efficient robot navigation [18].

2.2.4 Scenario Control and Oversight Mechanisms

For the overall scenario, the non-deterministic progress of the simulation must be specified, analogous to plot and story in a play, but not a single thread of narrative. For interactive theater, a scenario in the form of a partial ordering of scenes, in the form of a plot graph which is a PERT-chart-like directed acyclic graph defining a partial ordering of major events or scenes, is proposed by the Oz project group [10]. In this work, however, it is left unclear how the scenario form used would be handled automatically by software; the study published deals with experiments using human actors, and human understanding of the scenario.

The problem of scripting animations derived from simulation, and the similar problem of "experiment authoring" for driving simulations, are discussed by Booth et al [6] in the context of real-time simulation. However, the bulk of that discussion concerns the current state of their simulation techniques, and presentation of a planned new simulation engine, rather than what we mean by scenario (script in their terms).

They do however propose an interesting approach to those activities or events that should happen when a likely subject arrives in a certain area, or when the flow of events reaches a certain point. For this, Booth et al propose the idea of using what they call a *beacon*: the user places a beacon as a way of providing an invisible cue, at the desired time and place, for some action. This beacon can be seen as an agent which acts on behalf of the user, determining details needed to carry out the user's desires. In the driving simulator that this approach is used in, the desired action might be that a car fails to stop at a stop sign, thus potentially endangering the driving trainee. The timing of this event, and the selection of the offending vehicle, is dependent on which cars are in the vicinity and is not scriptable in any completely predetermined way. Hence the beacon coordinates things, selecting the car that is most appropriate to be the offender, and adjusting its behavior.

3 Sandia Virtual Environment Overview

The Sandia Virtual Environment allows multiple simulation engines to run concurrently on the local net, multiple viewing or display engines on the net, and other special purpose processes as needed. Simulation engines include:

1. The World Engine, which provides coordination of objects in the world, reasoning about world state, and simple physical simulation.
2. Figure simulation engines:
   (a) *Jack*®for *avatars* (the human figure models representing actual users)
   (b) Virtual Actor work in progress

Our current display engine, the VR Station, drives a range of output devices, including headmount displays and the Fakespace BOOM, as well as workstation monitors. It also processes input for viewing control.

Special purpose input gathering processes currently include:

1. Electro-magnetic tracker input
2. hand closure sensor input
3. mouse button input

Typically, the heavier weight components, i.e. the simulation engines and the VR Station, each run on a separate CPU, and in practice on separate machines, while auxiliary lighter weight processes share processors.

We have described the initial version of this system, with emphasis on the participants and their avatars, elsewhere [23].

4 Virtual Actor Research

The Virtual Actor work involves the construction of a testbed implementation, which will have two major aspects:

1. Simulation Engines for the Behavior of Individuals
   - For basic behaviors, these will draw on techniques from the literature, and possibly adapt existing software where appropriate. (Possible software sources include University of Pennsylvania's *Jack*®and Ken Perlin's techniques; the latter are, through collaborative arrangement, currently being tested.)
   - Group behavior will be a function of (new) higher-level behaviors implementing *roles* for individuals. One aspect of our research focus is that we want to provide a means of structuring the behavior of semiautonomous actors, while retaining reactive capabilities in these actors.
2. Scenario Support with Intervention Capability
   Another aspect of our research focus shows up here: structuring actor interactions in relation to the overall environment and the scene progression, while providing run-time control functionality for user oversight and intervention (high-level user guided).

4.1 Architecture

Our goal is to provide the appearance of a range of human behaviors with user direction possible, in a performance-oriented architecture. The planned architecture includes a hierarchy of component types, in which components will be loosely coupled, separate processes, with messages flowing up and down the hierarchy. Results of the simulation are multicast on the local net. The following is the hierarchy of component types:

1. The **User Interface Controller** deals with overall action possibilities and actor types: based on the *Scenario*, and the individual Virtual Actor's *Role*, and mediated through the puppet components described below.
2. The **Global Puppet Coordinator** maintains aspects of the world that are of concern to the Virtual Actors, and provides data about them in answer to actor queries.

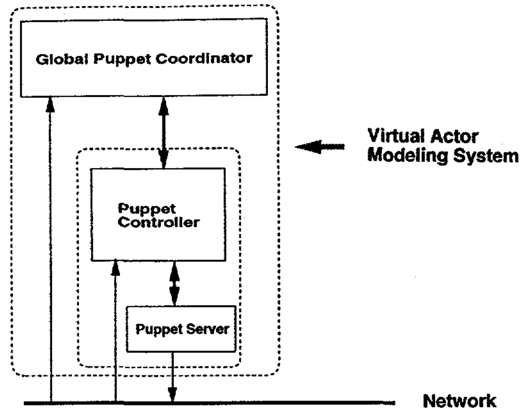

Figure 1: Complete Virtual Actor Components

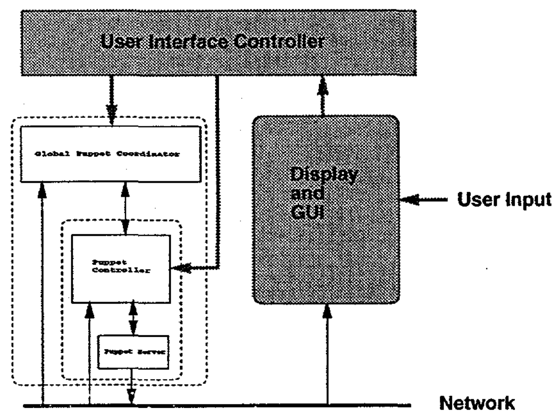

Figure 2: Virtual Actor Components in Context

3. The Virtual Actor's **Puppet Controller** component handles the individual Virtual Actor's constituent behaviors: implements Behavioral Procedures, Action Selection Mechanisms, and Steering Mechanisms.
4. The **Puppet Server**[2] handles the individual Virtual Actor's primitive motion capabilities: the *Motor Control Level*, largely or entirely kinematic, based on interpolation of driving functions, inverse kinematic support, and some general constraints.

The last three components altogether will comprise the Virtual Actor support system, see Figure 1. The last two components are (conceptually) replicated for each actor; they will comprise the Virtual Actor implementation per se. The overall architecture is placed in the context of the (eventual) user interface capabilities in Figure 2.

[2] term suggested by Perlin

4.2 Example Application Scenario

An example test application is an airport security scenario, which will include:

1. An airport setting comprising at least the following areas:
   (a) Walkway for boarding and deplaning passengers
   (b) Security checkpoint
   (c) Assorted shops, seating, and advertising kiosks
2. Extras of the following types:
   (a) Passengers moving through security checkpoint
   (b) Passengers milling about
3. A bit player of the following type:
   (a) Potential security checkpoint violator, who will provide multiple scenario outcomes:
      - He sets off the metal detector alarm and attempts to escape by running away.
      - He moves through the detector with no trouble.
      - He draws a gun and starts shooting, either before or after getting through the detector.
4. Loosely coupled interactions between simulated humans, allowing for some group or crowd behaviors, including perhaps background conversations (visual effect only – no attempt to simulate meaningful conversation).
5. The security guards who deal with the situation would be trainees, hence would appear as avatars in the scene.

4.3 Virtual Actor Current Status and Future Plans

We have the airport setting built, and are experimenting with a version of the puppet server derived from Ken Perlin's work, using a modified version of his dancer software running as a basic menu-driven puppet server. The figure can move about in the airport set, and work is proceeding on the puppet controller to provide more sophisticated behaviors. We will add the rest of the components of Section 4.1 after that.

We want eventually to have our Virtual Actors interact not only with each other, but with the human participants' avatars. This is a goal we will approach gradually, with the Virtual Actors being able, for now,

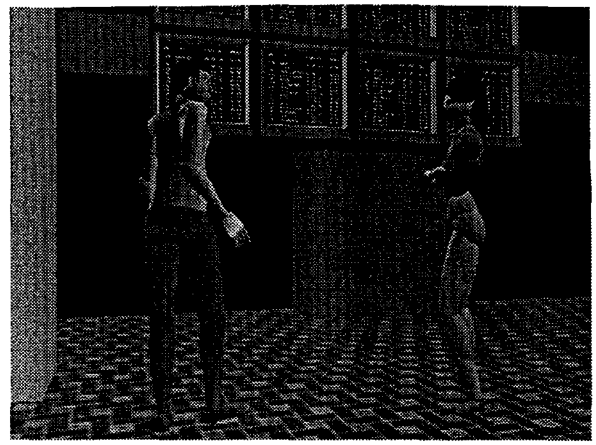

Figure 3: Virtual Actor and Avatar

to look at or apparently attend to an avatar, or to react if shot at, but we severely limit the recognition of avatar actions. See Figure 3 for a view of the Perlin dancer figure together with a *Jack*®-based avatar in our airport setting.

References

[1] Philip E. Agre and David Chapman. Pengi: An implementation of a theory of activity. *Cognitive Modeling*, pages 268–272, 1990?

[2] Philip E. Agre and David Chapman. What are plans for? In Pattie Maes, editor, *Designing Autonomous Agents*, pages 17–34. The MIT Press, 1990.

[3] Omar Ahmad, James Cremer, Joseph Kearney, Peter Willemsen, and Stuart Hansen. Hierarchical, concurrent state machines for behavior modeling and scenario control. May now be published, date assumed – got from net, 1994.

[4] Joseph Bates, Bryan Loyall, and W. Scott Reilly. Broad agents. *SIGART Bulletin*, 2(4):38–40, 1991.

[5] Welton Becket and Norman I. Badler. Integrated behavioral agent architecture. In *Recent Techniques in Human Modeling, Animation, and Rendering (SIGGRAPH 93 Course Notes 80)*, pages 4–39–4–50. 1993.

[6] Michael Booth, James Cremer, and Joseph Kearney. Scenario control for real-time driving simulation. In *Proceedings of the 4th Eurographics Workshop on Animation and Simulation*, pages 103–119, September 1993.

[7] Rodney A. Brooks. Elephants don't play chess. In Pattie Maes, editor, *Designing Autonomous Agents*, pages 3–15. The MIT Press, 1990.

[8] Rodney A. Brooks. A robot that walks: emergent behaviors from a carefully evolved network. In Norman I. Badler, Brian A. Barsky, and David Zeltzer, editors, *Making them move: mechanics, control, and animation of articulated figures*, pages 99–108. Morgan Kaufmann, 1991.

[9] Julie A. Dickerson and Bart Kosko. Virtual worlds as fuzzy cognitive maps. *Presence*, 3(2):173–189, 1994.

[10] Margaret Thomas Kelso, Peter Weyhrauch, and Joseph Bates. Dramatic presence. *Presence*, 2(1):1–15, 1993.

[11] A. Bryan Loyall and Joseph Bates. Real-time control of animated broad agents. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, pages 664–669, June 1993.

[12] Pattie Maes. Situated agents can have goals. In Pattie Maes, editor, *Designing Autonomous Agents*. The MIT Press, 1990.

[13] Pattie Maes, Trevor Darrell, Bruce Blumberg, and Alex Pentland. The ALIVE system: Full-body interaction with autonomous agents. *submitted to IEEE Computer Special Issue on Virtual Environments*, page ?, 1995.

[14] Howard Lee Mohn. Implementation of a tactical mission planner for command and control of computer generated forces in ModSAF. Master's thesis, Naval Postgraduate School, September 1994.

[15] Nils J. Nilsson. Reacting, planning and learning in an autonomous agent. DRAFT, June 1994.

[16] Nils J. Nilsson. Teleo-reactive programs for agent control. *Journal of Artificial Intelligence Research*, 1:139–158, January 1994.

[17] Hansrudi Noser, Olivier Renault, Daniel Thalmann, and Nadia Magnenat-Thalmann. Vision-based navigation for synthetic actors. In *Recent Techniques in Human Modeling, Animation, and Rendering (SIGGRAPH 93 Course Notes 80)*, pages 1–31–1–36. 1993.

[18] David W. Payton. Internalized plans: A representation for action resouces. In Pattie Maes, editor, *Designing Autonomous Agents*, pages 89–103. The MIT Press, 1990.

[19] Ken Perlin. A gesture synthesizer. In James F. Blinn, editor, *SIGGRAPH 94 Course Notes, Course 1*. 1994.

[20] Ken Perlin. Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics*, 1(1):5–15, March 1995.

[21] Mark W. Pfefferman. A prototype architecture for an automated scenario generation system for combat simulations. Master's thesis, Air Force Institute of Technology, 1994.

[22] Craig W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In Maureen C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 25–34, July 1987.

[23] Sharon Stansfield, Nadine Miner, Dan Shawver, and Dave Rogers. An application of shared virtual reality to situational training. In *VRAIS '95 Proceedings*, pages 156–161, 1995.

[24] Milind Tambe, W. Lewis Johnson, Randolph M. Jones, Frank Koss, John E. Laird, Paul S. Rosenbloom, and Karl Schwamb. Intelligent agents for interactive simulation environments. *AI Magazine*, 16(1):15–39, 1995.

[25] Xiaoyuan Tu and Demetri Terzopoulos. Artificial fishes: Physics, locomotion, perception, behavior. In *SIGGRAPH '94 Proceedings*, pages 43–50, July 1994.

[26] Toby Tyrrell. An evaluation of Maes's bottom-up mechanism for behavior selection. *Adaptive Behavior*, 2(4):307–348, 1994.

[27] David Zeltzer and Michael B. Johnson. Motor planning: An architecture for specifying and controlling the behaviour of virtual actors. *Journal of Visualization and Computer Animation*, 2:74–80, 1991.

[28] David Zeltzer and Michael B. Johnson. Virtual actors and virtual environments. In Lindsay MacDonald and John Vince, editors, *Interacting with Virtual Environments*, pages 229–255. John Wiley & Sons, Ltd., 1994.

[29] Michael J. Zyda, David R. Pratt, James G. Monahan, and Kalin P. Wilson. NPSNET: Constructing a 3d virtual world. In David Zeltzer, editor, *Computer Graphics (1992 Symposium on Interactive 3D Graphics)*, volume 25, pages 147–156, March 1992.