

INTELLIGENT OPPONENTS FOR VIRTUAL REALITY TRAINERS

Robert E. Wray
Soar Technology, Inc.
Ann Arbor, MI

John E. Laird
University of Michigan
Ann Arbor, MI

Andrew Nuxoll
University of Michigan
Ann Arbor, MI

Randolph M. Jones
Soar Technology, Inc.
Ann Arbor, MI

ABSTRACT

We are exploring the requirements for and prototypes of intelligent opponents for a virtual reality environment in which U.S. Marines can train to clear buildings of enemy soldiers. This virtual reality trainer is one of the components of the Virtual Training and Environments (VIRTE) Program of the Office of Naval Research. Autonomous, intelligent computer generated forces, such as TacAir-Soar, have been shown to be effective in vehicle-level entity domains. Creating models of individual humans engaged in combat with direct fire weapons presents a host of new challenges. One challenge for the current project is that intelligent agent development is concurrent with development of the simulation technology by other researchers, meaning the final virtual environment is not yet available for the agents. Our solution is to use a commercially available, first-person perspective shooter game (Unreal Tournament) as a prototyping environment. We describe requirements for intelligent opponents, our approach and the current system architecture. To date, we have developed agents that navigate through buildings and defend themselves. Creating much more realistic behaviors is a larger challenge and we introduce a number of potential research questions that could, if pursued, increase the realism of the VIRTE agents.

AUTHOR BIOGRAPHIES

Robert E. Wray is a Senior Scientist at Soar Technology. His research and experience encompass many areas of artificial research and include agent-based systems and agent architectures, machine learning, knowledge representation and ontology, and neural networks. Prior to joining Soar Technology, he held the position of assistant professor of computer science at Middle Tennessee State University. He received a Ph.D. in computer science and engineering from the University of Michigan in 1998. His doctoral research focused on maintaining logical consistency in agent reasoning systems and his innovations have been incorporated in the latest version of the Soar architecture.

John E. Laird is a Professor of Electrical Engineering and Computer Science at the University of Michigan and Associate Chair of the Computer Science and Engineering Division. He received his Ph.D. in Computer Science from Carnegie Mellon University in 1983. His research interests spring from a desire to understand the nature of the architecture underlying artificial and natural intelligence. He is one of the original developers of the Soar architecture and leads its continued development and evolution. From 1992-1997, he led the development of TacAir-Soar, a real-time expert system that simulates military pilots and is used for training in large-scale distributed simulations. Most recently his research has broadened to include creating human-like AI systems in computer games. He is a Fellow of AAAI and founder of Soar Technology, Inc.

Andrew M. Nuxoll is a graduate student in the Electrical Engineering and Computer Science department at the University of Michigan. He received his B.S. in Computer Science from Rose-Hulman Institute of Technology in 1995. He recently abandoned a promising six year career as a professional software engineer to pursue his Ph.D. He is enchanted by attempts to model intelligent behavior and hopes to make significant contributions to the AI field.

Randolph M. Jones received his B.S. in Mathematics and Computer Science from the University of California, Los Angeles, in 1984. In 1987 and 1989, respectively, he received M.S. and Ph.D. degrees in Information and Computer Science from the University of California, Irvine. He has held research positions at Carnegie Mellon University, the University of Pittsburgh, and the University of Michigan. He is currently Chief Scientist and Vice President of Soar Technology, Inc., and an assistant professor in the Computer Science Department at Colby College, where he is also department chair. His research interests lie in the areas of interactive intelligent systems, executable psychological models, and human and machine learning.

INTELLIGENT OPPONENTS FOR VIRTUAL REALITY TRAINERS

Robert E. Wray
Soar Technology, Inc.
Ann Arbor, MI

John E. Laird
University of Michigan
Ann Arbor, MI

Andrew Nuxoll
University of Michigan
Ann Arbor, MI

Randolph M. Jones
Soar Technology, Inc.
Ann Arbor, MI

INTRODUCTION

Offensive urban combat is one of the most difficult tasks soldiers are called upon to perform (USMC 1998). Urban combat is characterized by building-to-building and room-to-room fighting, and even a vastly outnumbered defensive force can inflict significant casualties on attacking forces. Effective, frequent training is an essential element in reducing casualties in urban combat. However, training in urban environments is costly and resource constrained (due, in part, to the terrain needed for training).

One of the goals of the Virtual Training & Environments (VIRTE) program, sponsored by the Office of Naval Research, is to develop new science and technology for immersive virtual trainers for urban combat. In this trainer, fire teams of four U.S. Marines will be situated in a virtual urban environment and tasked to clear a building, possibly containing enemy soldiers. Developing virtual trainers for urban combat requires investments and research in display technology, body tracking, haptics, human perception, and human behavior representation.

A critical component of the overall system is providing virtual, intelligent opponents for the soldiers participating in training in the simulation. This paper describes the general requirements for such intelligent opponents, our approach to developing intelligent opponents using the Soar cognitive architecture, our current software system, which employs a commercial computer game as a prototype development environment, and a discussion of the research issues in cognitive science and artificial intelligence that this domain presents.

REQUIREMENTS for INTELLIGENT OPPONENTS

An intelligent opponent should do more than provide a realistic experience for a Marine in the simulator. This section outlines some of the high level requirements for intelligent opponents in this domain.

1. **Observational fidelity:** For a training environment, it is critical that the simulation (including the intelligent opponents) provide a realistic experience as possible. However,

providing a realistic experience to trainees does not necessarily imply that the intelligent opponents need to be realistic in all possible aspects. What is crucial is that those aspects of a real opponent that are observable to a soldier be re-created in the virtual opponents. Behaviors that must be created include human-like reaction to light and noise, coordination and communication with other entities, courage and cowardice, indecision, and startle responses, as well as the doctrine and tactical knowledge appropriate for the opponents. However, it may not be necessary to create realistic models of, for example, startle responses, as long as the observed behavior to a startling event is congruent with human behavior.

2. **Faster than real-time performance:** Obviously, realistic behavior requires performance in real-time. However, latency is a critical issue in virtual environments and VIRTE is seeking to create a simulation on a single workstation personal computer. Thus, the intelligent opponents must be able to run much faster than real-time, as they will be given only a small time slice in the simulator cycle.
3. **Behavior variability:** Trainees must be exposed to many diverse experiences. In particular, to avoid negative training, VIRTE opponents should not be predictable: it is important to avoid creating a simulation that can be "gamed." For example, a player in a first-person perspective computer game may learn that enemies always appear in the same places in a room and can target those locations before enemies arrive. This behavior improves game play but does not transfer to the real world and reflect the unpredictability that actual human opponents would present. Diversity in actual human behavior arises from many sources, and attempts to model the VIRTE opponents should account for the sources of that diversity systematically, rather than arbitrarily adding "variability" parameters.
4. **Transparency:** Another important requirement is that an intelligent opponent be able to explain its actions. A planned component of the simulation is an "after action review," in which trainees can critique their performance. Being able to provide explanations of why opponents went into a particular room, or retreated at some juncture in the

scenario, requires some kind of explanation facility. Explanations are much easier to generate in systems that use transparent representations that map directly to understood terms.

5. **Rapid, low-cost development:** The intelligent opponents are only one of many components of the overall VIRTE urban combat trainer. Thus, the cost of the development of the intelligent opponents should be reduced as much as possible. This requirement is important because sophisticated intelligent computer generated forces (ICGFs) are being developed, but questions about the benefits of ICGFs, relative to their costs, remains an unresolved question in the military modeling and simulation community.

APPROACH

Intelligent computer generated forces have already been developed to support training and analysis simulations. ICGFs are embedded in realistic models of military vehicles, such as planes or ships. They are completely autonomous, making decisions based on their awareness of the surrounding environment, commands received from other entities, and their extensive knowledge of military doctrine and tactics. They have the ability to pursue mission objects alone, or they can participate in larger groups made up of other ICGFs or even human teammates participating via simulators. Our approach is to extend ICGF technology to the particular requirements of the physical representation of human bodies rather than vehicles (Reece, Ourston et al. 1998) and to address the requirements introduced in the previous section.

We have previously developed a number of ICGFs using Soar, a state-of-the-art artificial intelligence architecture. Soar is based on extensive research in building reactive and deliberate knowledge-based systems, as well as research on modeling the details of human behavior (Laird, Newell et al. 1987; Newell 1990; Wray and Laird 1998). Soar runs much faster than real-time on current workstations. Multiple agents have been used within a computer game with performance requirements comparable to those required in VIRTE (Laird, Assanie et al. 2002).

Soar agents will act as independent adversaries in a training scenario. To meet the realism requirements for the training application, an adversary must be able to perceive its environment and act appropriately. It must understand and negotiate its environment. When appropriate, it should act in a coordinated fashion with other members of its team. Soar agents have been successfully applied to tasks requiring each of these capabilities.

For example, TacAir-Soar is an ICGF Soar system that emulates the behavior of military personnel performing missions in fixed-wing aircraft (Tambe, Johnson et al. 1995; Jones, Laird et al. 1999). The goal in developing TacAir-Soar was to generate behavior that “looked human” and followed military doctrine. TacAir-Soar has demonstrated the ability to generate autonomous, real-time, high fidelity behavior for a large-scale simulation of a complete theater battle (Laird, Jones et al. 1998).

Soar is a symbolic architecture with explicit goals and data structures that can be readily understood by non-technical users. For example, a goal for the intelligent opponent might be to defend a room and a user could find a “defend-room” goal among the Soar agent’s currently active goals. Although this representation facilitates transparency, alone it does not achieve it because some knowledge of the run-time aspects of the architecture is needed to extract this information. However, we intend to use a tool that has been specifically developed to bridge this gap between the transparency of Soar’s representation and the extraction of this information from the agents themselves. The Visualization Toolkit for Agents (VISTA) extracts and displays agent state information automatically (Taylor, Jones et al. 2002), making it much easier for non-technical users to understand the decisions and behavior of Soar agents.

In addition to ICGFs, we have also developed individual adversaries for first-person perspective interactive computer games using Soar (Laird 2001). Although computer games are rarely appropriate for training, they serve as a useful environment for developing individual opponents with human-like behavior (Laird 2000). Further, the game opponents also require many of the capabilities required of opponents for the VIRTE simulator infantry training such as maneuvering through rooms, tracking targets, using spatial layout for tactical advantage, and anticipating enemy behavior (Laird 2001). In addition, we have been studying a number of behavioral parameters that are important for creating believable, human-like behavior (Laird and Duchi 2001). These parameters will be critical in allowing us to realize variability in opponent behaviors.

Our experience developing ICGFs and computer games allows us to leverage methodologies, designs, tools, and implementations for the VIRTE intelligent opponents and thus reduce the overall development cost. For example, the development technology for TacAir-Soar has transferred to a number of similar applications including the emulation of rotary-wing aircraft behaviors (Hill, Chen et al. 1997), among others.

In a few cases, data structures and agent knowledge will transfer directly to the new system. For example, the communications infrastructure for TacAir-Soar has been intentionally designed as a reusable ICGF component (Wray, Beisaw et al. 2002). This communications knowledge should transfer directly to the VIRTE effort with little modification other than customization for the different types of communication in this domain. Further, as described above, we do expect some aspects of the implementations of computer game opponents to be applicable for the VIRTE opponents. In general, however, little of the specific knowledge used in these systems will transfer directly to VIRTE adversaries.

PROTOTYPE SOFTWARE ARCHITECTURE

Part of the overall VIRTE program includes the development of a state-of-the-art simulation system that will integrate all the components of the virtual environment. As that system is under development, we need a prototyping and development environment that will allow us to pursue the development of the intelligent opponents. Our solution is to use a commercially available, first-person perspective shooter game, Unreal Tournament, for a prototyping environment. Unreal Tournament (UT) provides an off-the-shelf, high-quality 3D game engine that can be easily extended. All of the game physics and interface

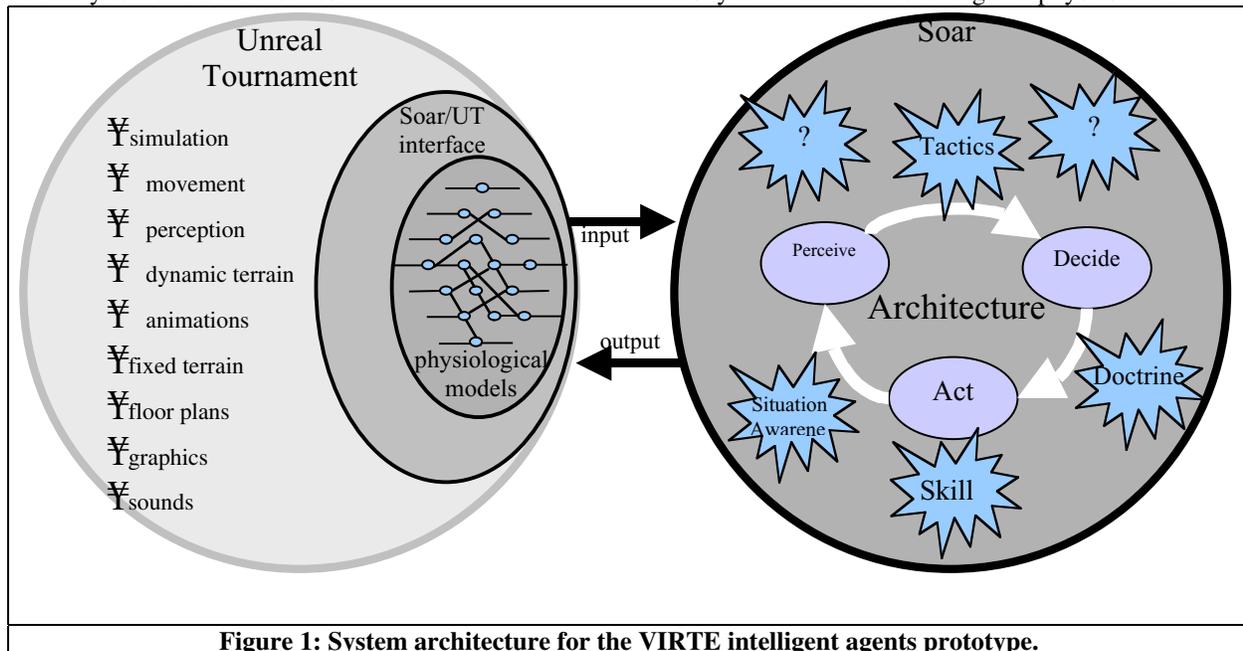


Figure 1: System architecture for the VIRTE intelligent agents prototype.

Although direct agent knowledge transfer will be limited, the development methodologies and many component and data structure designs will be reused throughout the design and development of the intelligent opponents. The development of these systems has also provided us with rich tools that support creating and deploying large-scale, knowledge-based systems. Together, the reuse of methodologies, tools, designs, and, in some cases, agent knowledge, makes a low-cost implementation of complex, interactive intelligent opponents more feasible.

Another group of developers is using a similar system architecture but a different AI architecture for this domain (Best, Lebiere et al. 2002). Both groups are using the same resources for Marine doctrine and training (USMC 1998) and the same subject matter experts for knowledge acquisition and simulation.

is coded in a powerful internal scripting language (Unrealscript) that is completely accessible. The game is inexpensive (\$20 US) and flexible and an active user community provides many of the game-specific tools and graphics needed for scenario generation. Leveraging the tools and experience of a much larger user base allows us to focus the majority of our efforts on developing the opponent behaviors. If we are deliberate in our designs, the behaviors we develop for UT should readily transfer to the VIRTE training system when it is built.

A good example of the flexibility and extensibility of UT, and its ambitious and creative user community, is the game modification "Infiltration." Unreal Tournament is set in a future world with futuristic weapons (lasers) and combatants clad in space-suit-like uniforms.

Infiltration is a freely available extension of Unreal Tournament that provides graphics and models of modern weapons (e.g., M-16 rifles), more sophisticated models of the impact of wounds, and representations of modern army uniforms and equipment. Because we are using Infiltration as our development environment, we will not need to focus significant energy towards creating realistic representations of modern combatants in Unreal Tournament.

Figure 1 presents our system architecture. The cognitive aspects of the intelligent opponent are run in Soar. Soar uses input from the simulation and knowledge (such as doctrine, situation interpretation knowledge, skill knowledge, and mission-specific knowledge) to generate actions that are communicated to Unreal Tournament. Unreal Tournament and Infiltration provide the simulation environment, including the representation of the environment (floor plans of buildings), animations, sounds and the execution of agent movement and creation of agent perception. Implemented within UT (in Unrealscript) is a domain specific module that translates UT data structures to Soar percepts and Soar actions to UT function calls. The physiological component within the Soar/Unreal Tournament interface will be described below.

In order to act as a human might act, the intelligent opponent agents must be well grounded in their environment. We believe that the best approach for realizing human-capable virtual agents is to provide them with realistic sensing and actions in their environments. Human perception can be adversely affected by many factors, including the level of light, smoke, and moving objects. As demanded by observational fidelity requirement, we must determine which of these (and those in other sensor modalities) must be modeled so that the automated adversaries have appropriately realistic behavior (and also which ones can be ignored). These perceptual models will be implemented in the prototype in the Soar/UT interface. Similarly, the interface must support the actions of the adversaries, such as crouching, kneeling, running, etc.

Realistic perception is challenging because of the difficulty of sensing walls (and doors) in simulation environments. Thus, initially we are introducing a few simplifications. Each agent receives an annotated map with regions that give the name of each room. This allows an agent to sense in which room it is located. Navigation nodes are placed in the map at important locations (doors, windows). The opponents will use these nodes for navigation between rooms, but can move freely within a room. The opponents move using controls similar to those used by a human game player (turning left and right; thrusting forward, backward,

left, and right). This style of control is more challenging for behavior development than directly moving to nodes or objects, but provides more flexibility in controlling the character during the scenario.

Grounding agents in their environment requires physiological models that can be used to influence behavior. Physiological variables may include temperature, fatigue, and hunger, among many others. Models of agent physiology are being developed for this domain and a role-playing computer game called *Haunt2*, also implemented in this software environment (Laird, Assanie et al. 2002). UT has been extended so that agents have a model of physiological responses to the environment and to their internal processing, as illustrated notionally in Figure 1. The environment has been extended so that it has attributes that influence the physiology of the characters. The physiological properties serve as input into the intelligent opponents; for example, the agent is aware of its temperature and level of fatigue. Changes in the attributes can affect others, so that a significant drop in body temperature can increase fatigue. Physiological effects that have been implemented include: temperature, exertion, fatigue, sleepiness, hunger, and thirst (Laird, Assanie et al. 2002). However, we have not yet determined which of these factors will be appropriate in the VIRTE application.

The functional role of the physical drives in the intelligent opponent system is to provide increased realism and contribute to behavior variability. Our current plan is to develop a means for physiological impacts to influence knowledge directly, rather than via modifications to the processing of the architecture itself, which we have previously explored when modeling fatigue (Jones, Neville et al. 1998). Regardless of whether the physiological factors influence the architecture directly or are realized via knowledge, they will push our agent technology to integrate knowledge-based, goal-oriented reasoning that has characterized most of our work with ICGFs and computer game opponents to date, with emotions, personality, and physical drives that have been used in simple, knowledge-lean agents in other systems (Loyall and Bates 1997; Macedonia 2000).

The final remaining component of the software architecture is the interface connecting Soar and Unreal Tournament. We are using a flexible, reusable C++ interface library called the Soar General Input/Output Interface (SGIO). SGIO is a high-level interface that is really three low-level interfaces: a socket-based interface that can be connected across separate computers, a socket-based interface between multiple processes on the same computer, and a C-language-based interface that connects the components within a

single process. The user can select from among the three low-level interfaces at run time without making any changes to the game engine or Soar. This configurability allows a developer access to development and run-time tools during the agent development process while allowing the most efficient, within-process interface to be used in the embedded application. SGIO will also be appropriate for interfacing to the VIRTE urban combat simulator.

RESEARCH DIRECTIONS

We have addressed the approach and tools we are using to develop intelligent opponents for the VIRTE urban combat training environment. Achieving behavior models for complex tasks that meet the demands of observational fidelity is a significant task. Our approach to achieving behavioral realism starts with trying to create *competent* synthetic entities. Much of human behavior is determined by the constraints of the task environment, and it is essential to create an initial system that at least can perform the basic tasks of attacking and defending, using knowledge of doctrine and tactics. Based on our experience with TacAir-Soar and our computer game characters, we expect to be able to create intelligent opponents that are *competent* at some subset of the task quickly. Our goal is to model behavior at the time scale above 500 ms, meaning that we do not intend to model all the low-level memory and perception details of the opponents. We expect that for training applications, this level of behavior is appropriate for the demands of observation fidelity. For some specific behaviors, we might find that more precise modeling is necessary, and principled, low-level perception and action has been previously modeled in our framework (Chong and Laird 1997).

In creating the initial competent synthetic adversaries, we are making simplifications we know do not accurately model the underlying processing or sensing used by humans (e.g., the map annotations). As we develop competent synthetic entities, we can identify where they fall short of realistic human behavior, and where those shortcomings have an impact on the training application. We may determine that some of the shortcuts, while allowing the synthetic characters to avoid some processing done by humans, are beyond the state of the art to model accurately and do not significantly impact the training value of the entities.

Thus far, in our focus on competent opponents, we have developed individual opponents that will defend a room and retreat when faced with sufficient firepower. Our development will continue, adding more opponent interaction (real coordination, communication, and teamwork) and deeper knowledge of tactics, doctrine,

and situation assessment. However, we are also beginning to recognize some of the simplifications and shortcuts that will need to be explored more fully. In the remainder of this section, we outline four research areas potentially important to this application: emotions, teamwork, spatial reasoning, and dynamic direction. We focus on the behavior requirements that make these questions relevant for this domain.

Emotions

In addition to using physiological models to introduce behavior variability, we also plan to incorporate research on building appropriate emotional responses into intelligent agents (Henninger, Jones et al. 2001; Jones, Chown et al. 2001; Chown, Jones et al. 2002; Jones, Henninger et al. 2002). This computational model of emotions has been developed in a demonstration framework that included special operations forces performing a simple reconnaissance mission. Enough similarity exists in approach and design between the special operations application and the intelligent opponents to map the work relatively directly to the VIRTE agents.

Because the theory underpinning this model assumes that emotional responses were ultimately provided by evolution, the model assumes that these constraints are, on average, beneficial to decision making. However, it is clear that not all emotional responses are always beneficial. Because we want to introduce principled variability in decision making, the impact of emotions may have the desired effect, providing a mix of normal, "smart" behavior and occasionally overriding "emotional" behaviors. Understanding and evaluating the role of this emotional model in the context of the intelligent opponents is a research question itself, and will also likely drive additional refinements to the existing model.

Coordination & Teamwork

Teamwork is essential to the operation of military forces, from infantry fire teams to strategic planning. A reusable, task-independent model of teamwork, STEAM, implemented in Soar, was created to demonstrate an effective way for agents to perform in a collaborative, multi-agent environment (Tambe 1997). STEAM provides agents the ability to reason about team-specific goals, and to respond appropriately when team-impacting events occur (e.g., loss of a platoon leader). This flexibility contrasts to many AI systems, in which team behavior is hard-coded, providing little direction for team coordination when unexpected events occur. STEAM has been deployed in a real-time

simulation of a soccer team and in RWA-Soar, among others

The STEAM model of teamwork is based upon the concept of a *joint intention* (Cohen and Levesque 1991) in which all members of a team intend to perform some action together. Once this joint intention has been established, the team has created a *joint persistent goal* and the task is undertaken. In STEAM, certain actions are designated as *team operators* that must be performed as a joint persistent goal. Once a team member realizes that conditions are met for undertaking a team operator, it announces this fact to the team. Each team member then verifies that it is committed to performing that action. Once all team members are committed, a joint persistent goal has been established and the action is performed.

We intend to incorporate elements of the STEAM model into the adversaries to form a basis for team representations. One of the challenges in using STEAM is that it depends on shared team goals to understand what other team members are doing. In a domain where military doctrine or rules (as in soccer) dictate a relatively small range of goals, recognizing another team member's goal may not be as difficult. However, in the relatively unconstrained environment of infantry engagement, recognizing team goals from an individual's actions will be difficult. STEAM also requires *a priori* enumeration of possible teams. As team creation and dissolution is much more dynamic in the urban combat domain, we will need to develop solutions to relax this constraint. Whether these limitations lead us to extend STEAM or simply to use it as inspiration for our own, specialized team models is an open question of investigation.

Spatial Reasoning

The intelligent opponents will need to perceive and understand the spatial layout of their surrounding environment: where are the doors, to what rooms is a hallway connected, what exterior areas can be viewed from a window while remaining concealed, etc. In order to create realistic synthetic adversaries, we will have to incorporate tactical spatial perception and reasoning that understands both structural (doors, rooms, windows) and tactical (concealment) aspects of the spatial environment. The ultimate goal is to have synthetic entities that can be "dropped" into any virtual environment and in real time build up their own understanding of the world without human assistance. However, that goal is extremely difficult to achieve today. As we outlined previously, our approach is to provide much of the important structural information *a priori*, via map annotation. We will also manually

identify tactically important locations, such as sniper positions. This annotation will allow us to create competent agents quickly. However, we plan to explore a series of extensions that would make the opponents increasingly less reliant on annotation. An obvious first step is to create tools that perform the annotations automatically, similar to the terrain database labeling that has been automated in other simulation systems (Reece, Kraus et al. 2000). The intelligent opponent agent currently constructs a wall-and-door map of the environment as it gains experience in its environment, so another key step would be adding knowledge to allow it to recognize and label rooms, hallways, etc. However, this problem has currently not been solved in a general way by researchers in artificial intelligence or robotics.

Dynamic Direction

Doctrine and tactics alone do not dictate the behavior of adversaries in a training exercise. Behavior is also dictated by the training goals of exercise (e.g., one goal might be to improve trainees' abilities in situational assessment). Human trainers at existing urban combat training sites often create behavior scripts for human soldiers playing the role of opponents in training exercises. These scripts specify a timeline of behavior of the adversaries at a high level – when specific events should occur, where adversaries should be located, what their goals should be. To create effective training situations, we need to give the intelligent opponents the ability to both follow a high level script and also improvise in a context dependent way (e.g., if a script indicated to move to a location at a certain time, and at that time that location was held by the opposing forces, the agent should develop an alternate plan). For this first step, we need to develop new representations that will allow our agents to balance direction with believability of behavior and there has been some progress in this area (Assanie 2002).

Behavior scripts are necessarily static. Imagine a situation in which some trainees were observed to be ignoring closets as they cleared the rooms in a building. To maximize the training experience, the trainees should be presented with some situations in which intelligent opponents have secreted themselves in closets. This situation is much more dynamic and requires a director that monitors the progress of the scenario and directs the opponents during execution to perform actions that increase training effectiveness. An obvious research issue is that agents must be able to accept active direction and incorporate that direction coherently with their current plans. Moreover, we want to investigate the development of autonomous scenario directors that monitor the training exercise and can

make dynamic modifications to the script in order to optimize the achievement of training goals (Magerko 2002).

CONCLUSION

We have reviewed the requirements for intelligent opponents in a virtual environment for urban combat training, our approach to modeling behavior using the Soar cognitive architecture, our use of an existing computer game as a prototyping environment, and some initial research directions that have been identified as a consequence of our initial experiences in this domain. Undoubtedly, achieving intelligent opponents that are fully autonomous, believable, interactive, and directable, while exhibiting a variety of behaviors in similar circumstances, will require much additional research and development. However, we believe our approach is particularly suited to this challenge, given success in previous, similar efforts, and continuing advances in the agent architecture and development methodologies and tools. Further, previous work also provides, methodologies, tools, agent designs and agent code we can reuse in pursuit of the development of intelligent opponents for virtual reality training.

REFERENCES

- Assanie, M. (2002). Directable Synthetic Characters. AAAI 2002 Spring Symposium Series: Artificial Intelligence and Interactive Entertainment.
- Best, B. J., C. Lebiere, et al. (2002). Modeling Synthetic Opponents in MOUT Training Simulations Using the ACT-R Cognitive Architecture. Eleventh Conference on Computer Generated Forces and Behavior Representation, Orlando, Florida, Institute for Simulation and Training, University of Central Florida.
- Chong, R. S. and J. E. Laird (1997). Identifying dual-task executive process knowledge using EPIC-Soar. 19th Annual Conference of the Cognitive Science Society, Lawrence Erlbaum.
- Chown, E., R. M. Jones, et al. (2002). An Architecture for Modulating Agent Decision Making Through Emotions. Sixth International Conference on Autonomous Agents, Bologna, Italy.
- Cohen, P. R. and H. J. Levesque (1991). "Teamwork." Nous 35.
- Henninger, A. E., R. M. Jones, et al. (2001). A Symbolic-Connectionist Framework for Representing Emotions in Computer Generated Forces. Interservice/Industry Training/Simulation and Education Conference, Orlando, Florida.
- Hill, R. W., J. Chen, et al. (1997). Intelligent Agents for the Synthetic Battlefield: A Company of Rotary Wing Aircraft. Ninth Conference on Innovative Applications of Artificial Intelligence, AAAI Press.
- Jones, R. M., E. Chown, et al. (2001). A Hybrid Symbolic-Connectionist Approach to Modeling Emotions. Emotional and Intelligent Agents II: The Tangled Knot of Social Cognition, AAAI Press.
- Jones, R. M., A. E. Henninger, et al. (2002). Interfacing Behavior Moderators with Intelligent Synthetic Forces. Eleventh Conference on Computer Generated Forces and Behavior Representation, Orlando, Florida.
- Jones, R. M., J. E. Laird, et al. (1999). "Automated Intelligent Pilots for Combat Flight Simulation." AI Magazine 20(1): 27-42.
- Jones, R. M., K. Neville, et al. (1998). Modeling Pilot Fatigue with a Synthetic Behavior Model. Seventh Conference on Computer Generated Forces and Behavioral Representation, Orlando, Florida.
- Laird, J. E. (2000). An Exploration into Computer Games and Computer Generated Forces. Ninth Conference on Computer Generated Forces and Behavioral Representation, Orlando, FL.
- Laird, J. E. (2001). It Knows What You're Going To Do: Adding Anticipation to a Quakebot. Fifth International Conference on Autonomous Agents (Agents 2001), Montreal, Canada.
- Laird, J. E. (2001). "Using a Computer Game to Develop Advanced AI." Computer 34(7): 70-75.
- Laird, J. E., M. Assanie, et al. (2002). A Test Bed for Developing Intelligent Synthetic Characters. AAAI 2002 Spring Symposium Series: Artificial Intelligence and Interactive Entertainment.
- Laird, J. E. and J. C. Duchi (2001). Creating Human-like Synthetic Characters with Multiple Skill Levels: A Case Study using the Soar Quakebot. AAAI 2001 Spring Symposium Series: Artificial Intelligence and Interactive Entertainment, AAAI Press.
- Laird, J. E., R. M. Jones, et al. (1998). Lessons learned from TacAir-Soar in STOW-97. Seventh Conference on Computer Generated Forces and Behavioral Representation, Orlando, Florida.
- Laird, J. E., A. Newell, et al. (1987). "Soar: An architecture for general intelligence." Artificial Intelligence 33(3): 1-64.
- Loyall, A. B. and J. Bates (1997). Personality-Rich Believable Agents That Use Language. First International Conference on Autonomous Agents, Marina del Rey, California.
- Macedonia, M. (2000). "Using Technology and Innovation to Simulate Daily Life." Computer: 110-112.
- Magerko, B. (2002). A Proposal for an Interactive Drama Architecture. AAAI 2002 Spring Symposium Series: Artificial Intelligence and Interactive Entertainment.

- Newell, A. (1990). Unified Theories of Cognition. Cambridge, Massachusetts, Harvard University Press.
- Reece, D., M. Kraus, et al. (2000). Tactical Movement Planning for Individual Combatants. Ninth Conference on Computer Generated Forces and Human Behavior Representation, Orlando, Florida.
- Reece, D., D. Ourston, et al. (1998). Updating ModSAF for Individual Combatants: The DI SAF Program. 7th Conference on Computer Generated Forces and Behavior Representation, Orlando, Florida, Institute for Simulation and Training, University of Central Florida.
- Tambe, M. (1997). "Towards Flexible Teamwork." Journal of Artificial Intelligence Research (JAIR) 7: 83-124.
- Tambe, M., W. L. Johnson, et al. (1995). "Intelligent Agents for Interactive Simulation Environments." AI Magazine 16(1): 15-39.
- Taylor, G., R. M. Jones, et al. (2002). VISTA: A Generic Toolkit for Agent Visualization. Eleventh Conference on Computer Generated Forces and Behavioral Representation, Orlando, Florida, University of Central Florida.
- USMC (1998). Marine Corps Warfighting Publication (MCWP) 3-35.3: Military Operations on Urbanized Terrain (MOUT), United States Marine Corps.
- Wray, R. E., J. C. Beisaw, et al. (2002). General, maintainable, extensible communications for computer generated forces. Eleventh Conference on Computer Generated Forces and Behavioral Representation, Orlando, Florida, Institute for Simulation and Training.
- Wray, R. E. and J. E. Laird (1998). Maintaining consistency in hierarchical reasoning. Fifteenth National Conference on Artificial Intelligence, Madison, Wisconsin.