NPS-MV-02-002

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Innovations in Computer Generated Autonomy at the MOVES Institute

by

John Hiles
Michael VanPutte
Brian Osborn
Michael Zyda

December 2001

Approved for public release; distribution is unlimited.

Prepared by: MOVES Institute

NAVAL POSTGRADUATE SCHOOL
Monterey, California 93943-5000

RADM David R. Ellison, USN
Superintendent

Richard Elster
Provost

This report was prepared for and funded by the MOVES Institute.

This report was prepared by:

John Hiles
Professor

Brian Osborn
Commander, USN

Michael VanPutte
Major, USA

Michael Zyda
Director, MOVES Institute

Reviewed by:

Michael Zyda
Director, MOVES Institute

Released by:

D. W. Netzer
Associate Provost and
Dean of Research

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form approved OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)	**2. REPORT DATE** 17 December 2001	**3. REPORT TYPE AND DATES COVERED** Technical Report

4. TITLE AND SUBTITLE Innovations in Computer Generated Autonomy at the MOVES Institute	**5. FUNDING**
6. AUTHOR(S) Professor John Hiles, Major Michael VanPutte, Commander Brian Osborn, Dr. Michael Zyda	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Modeling, Virtual Reality and Simulations Institute, Naval Postgraduate School Monterey, California, 93940	**8. PERFORMING ORGANIZATION REPORT NUMBER** MOVES Institute, NPS-MV-02-002
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	**12b. DISTRIBUTION CODE**

13. ABSTRACT (Maximum 200 words.)

The MOVES Institute's Computer-Generated Autonomy Group has focused on a research goal of modeling intensely complex and adaptive behavior while at the same time making the behavior far easier to create and control. This research has led to five new techniques for agent construction, which include a social and organization relationship management engine, a composite agent architecture, an agent goal apparatus, a structure for capturing and applying procedural knowledge (*tickets*), and the ability to bring these technologies to bear at the right time and in the proper context through *connectors*.

The MOVES Institute, located on the campus of the Naval Postgraduate School, specializes in Department of Defense related research and applications, including projects in 3D visual simulation, networked virtual environments, computer-generated autonomy, human performance engineering, technologies for immersion, evolving operational modeling and defense/entertainment collaboration [http://movesinstitute.org/]. This paper provides a high level overview of the technologies developed by the Computer-Generated Autonomy Group including a description of research projects in the area of helicopter test and evaluation program planning, land navigation route planning, modeling the effects of organizational changes on infantry units, integrating autonomous agents into networked virtual environments, generating interactive stories, and modeling computer security.

14. SUBJECT TERMS Agents, multi-agent systems, autonomous systems, complex-adaptive systems, story-telling	**15. NUMBER OF PAGES** 41
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	ul

NSN 7540-01-280-5800

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The MOVES Institute's Computer-Generated Autonomy Group has focused on a research goal of modeling intensely complex and adaptive behavior while at the same time making the behavior far easier to create and control. This research has led to five new techniques for agent construction, which include a social and organization relationship management engine, a composite agent architecture, an agent goal apparatus, a structure for capturing and applying procedural knowledge (*tickets*), and the ability to bring these technologies to bear at the right time and in the proper context through *connectors*.

The MOVES Institute, located on the campus of the Naval Postgraduate School, specializes in Department of Defense related research and applications, including projects in 3D visual simulation, networked virtual environments, computer-generated autonomy, human performance engineering, technologies for immersion, evolving operational modeling and defense/entertainment collaboration [http://movesinstitute.org/]. This paper provides a high level overview of the technologies developed by the Computer-Generated Autonomy Group including a description of research projects in the area of helicopter test and evaluation program planning, land navigation route planning, modeling the effects of organizational changes on infantry units, integrating autonomous agents into networked virtual environments, generating interactive stories, and modeling computer security.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

In 1997, the National Research Council issued a report that specified a joint research agenda for defense and entertainment modeling and simulation [NRC, 97]. The research areas identified by that report highlighted the need for a non-traditional degree program that focuses more closely on issues specific to immersive technology and computer generated autonomy. The NRC report provides a guide as to what research and development is needed to develop our future interactive entertainment modeling and simulation systems. As a consequence of that report, a number of research laboratories have developed a joint entertainment/virtual reality, entertainment/defense or entertainment/NASA focus. The Naval Postgraduate School MOVES (Modeling, Virtual Environment, and Simulation) Institute, with the largest modeling, virtual environments and simulation academic degree program, is one such organization following that report's research agenda with a number of active projects in defense and defense/entertainment collaboration.

MOVES initial focus was grounded in a decade of work by the NPSNET Research Group in the area of networked virtual environments (net-VEs). This group focused on human-computer interaction and software technology for implementing large-scale virtual environments. As net-VEs continued to develop, the need for autonomous computer-generated characters capable of interacting intelligently with the participants continued to grow.

In 1999, MOVES added a new research direction in the area of multi-agent systems and computer generated autonomous behavior. From the outset, MOVES agent research has had two goals. First, to bring rich, complex, adaptive behavior to Department of Defense (DoD) related models, simulations and other systems through the application of multi-agent technology. And second, to make this adaptive behavior far easier to achieve and control. This latter characteristic will allow problem solvers to focus their attention and intellect on the agent's problem solving behavior and not on the implementation mechanism. The intent is to shift the focus away from "how do we do this?" to "what can we do with this?".

The first formal course in computer-generated autonomy (MV-4015 Agent-Based Autonomous Behavior for Simulations) was introduced to the MOVES curriculum in January 2000. Two years and four classes later, more than a dozen Masters theses have been published in this area, and the MOVES Computer-Generated Autonomy Group is on its third generation agent architecture.

This paper describes the motivation for performing defense related multi-agent research, explores previous and ongoing multi-agent system (MAS) simulation research projects within the MOVES Institute, describes several new and exciting innovations in the field of agent-based system simulation, and provides a roadmap for where MAS research at the MOVES Institute is headed.

II. SEMI-FLUID SOFTWARE STRUCTURE AND EMERGENT BEHAVIOR

A. INTRODUCTION

Software development has traditionally focused on building software based on rigidly structured architectures with terms like “structure” and “architecture” usually referring to fixed and immutable relationships among the components inside the software. Many in the computer science and software engineering (SE) community assume structure must be rigid and tightly bound at design time if a program has any chance of meeting its design goals. This outlook is analogous to our view of a new highway system that is designed on paper and constructed with concrete and steel to meet the forecast needs of a growing city. Once built, the highway system remains fixed and static unless new construction occurs. It would be absurd to expect it to mold itself into new forms to meet growing infrastructure and changing traffic patterns. This same thinking has held true for traditional software designs. The architecture is fixed at design time; its structure is inert.

The study of computer generated autonomous behavior is supplementing this thinking by exploring the use of multi-agent systems (MAS) to build software that modifies its own structure, within a set of constraints, to maintain close contact with a dynamic environment. MAS research at the MOVES Institute is founded on the premise that semi-fluid software structures are not only possible, but essential to developing truly adaptive simulations and modeling emergent behavior.

B. A DESIGN PARADIGM SHIFT

A real challenge when first encountering multi-agent system simulations is coming to grips with emergent behavior in software. Most software developers and programmers have been trained in traditional software engineering, relying on rigidly structured system designs that implement a *direct* solution to the problem. Traditional problem solving in software engineering is *direct* in the sense that the developer conceives of an algorithmic solution and transfers that solution to software. Software development rigor and practice is used to insure the code will produce an exact execution of the algorithm. In direct solutions, the programmer knows exactly how to solve the problem and the software implements that solution precisely. This approach is fine for problems where the domain is well know, and the relationships are static, finite and well defined. Direct solution systems are somewhat analogous to well-behaved functions. For a given input, the designer knows what to expect for the output. Surprises are a clear indication of a bug in the system.

In sharp contrast, surprises in MAS simulations are not only okay, but are the desired end, as long as the system operates within boundaries that are explicitly determined. The software is intended to surprise the designer within a system of constraints! This is possible through the use of software agents that discover an *indirect* path to the solution, thereby allowing for the possibility of arriving at a solution the designer may not have previously considered. In this way, multi-agent systems are capable of producing innovative solutions. These solutions are *indirect* in that they were not explicitly programmed into the software; rather they are solutions that are consistent within the constraints the designer places on the software agents. As a result, any solution that is valid within the imposed constraints, is no longer a bug, but a potential novel approach to the problem.

Learning to design and implement software capable of emergent behavior, as well as recognizing the difference between “emergent behavior” and a “bug”, is the first step to developing complex agent-based simulations. One of the authors has taught the MOVES courses on computer-generated autonomy since their inception in January 2000 [Hiles, 1999]. The introductory course builds on three principal problems (and their solutions):

- Brian Arthur’s El Farol Bar problem [Arthur, 1994] serves as an introduction to the use of inductive thinking and indirect solutions;
- Boids by Craig Reynolds [Reynolds, 1987] explores the possibilities of autonomous control and self-organizing groups of problem solving vehicles (hardware or software);
- Andrew Ilachinski’s ISAAC: An Artificial-Life Approach to Land Warfare [Ilachinski, 1997] introduces the complexity of social behavior and relationships.

The first generation of multi-agent simulation projects that emerged from the computer-generated autonomy course were relatively simple. With no prior work to build upon, most of the thesis work was devoted to designing an architecture and very little time was spent understanding the behavior of the system.

A major step forward within the MOVES Computer-Generated Autonomy Group occurred with the introduction of the RELATE architecture. RELATE is an agent architecture for organizing agents into relationships, and allowing for functional specialization [Roddy and Dickson, 2000]. Once complete, the architecture simplified the construction of a variety of agent simulations including the dynamic exploration of helicopter reconnaissance [Unrath, 2000] and modeling tactical level combat [Pawloski, 2001]. These models represented the second-generation of work and provided a springboard for implementing models with greater complexity and richer behavior.

With a solid foundation of MAS models to build upon, students were able to take abstract concepts, and move more quickly from design to implementation. The focus of the third-generation work moved from engineering and deduction to a more inductive approach and even a hybrid approach using data generated from synthetic laboratories to gain insight into real world problems [Ercetin, 2000], [French, 2000].

Most recently, the MOVES Computer-Generated Autonomy Group has taken advantage of some real innovation in agent technology introduced by John Hiles to greatly simplify the creation of far more complex behavior. These innovative ideas have been put to the test in a new round of thesis projects [Mert and Jilson, 2001], [Hennings, 2001], [Washington, 2001] and ongoing research projects in the area of computer security, interactive stories, and auto-narration of agent-based simulations.

III. INNOVATIONS IN AGENT RESEARCH

Progress at the MOVES Institute over the past three years has been very exciting. The Computer-Generated Autonomy Group has developed five key technologies that significantly further the research goal of making far more complex and adaptive behavior easer to create and control. The key technologies include a social and organizational relationship management engine, a composite agent architecture, an agent goal apparatus, a structure for capturing and applying procedural knowledge (*tickets*), and the ability to bring these technologies to bear at the right time and in the proper context through *connectors*.

A. SOCIAL AND ORGANIZATIONAL RELATIONSHIP MANAGEMENT ENGINE

The modeling and simulation community is continually being challenged to create rich, detailed models of ill-defined problems. Many of these problems are complex because of the involvement of human decision-making and organizational behavior. Humans and organizations have multiple levels of internal roles, goals and responsibilities, frequently conflicting with each other. While contemplating almost any decision, humans must evaluate a myriad of goals that they are currently attempting to achieve. These goals are sometimes supportive of each other, but often they are in conflict. Developing simulations that are capable of capturing this complex, often unpredictable, behavior is essential to realistically modeling large organizations accurately.

In an effort to simplify the development of MAS simulations and ease the integration of software agents into existing simulations, an agent modeling architecture called RELATE was created [Roddy and Dickson, 2000]. The RELATE design paradigm proposes an effective way to model the complex, human decision-making process that focuses on how an individual relates to other things and individuals within its environment. By concentrating on the relationships of individuals and within organizations, the developer is encouraged to identify the various roles that are assumed by members belonging to each relationship. These roles have certain responsibilities and commitments, which tend to be manifested as additional goals that must be addressed by the various members of the relationship. Once an agent is a member of a relationship, it must base its action selection on its personality, its particular concern for each goal, and the state of achievement of each goal. Entering into a relationship connects or binds agents to one another, resulting in the assignment of new roles, goals and responsibilities. Relationships are often formed to achieve something that is not achievable by any one individual. In this way, agents can take advantage of shared resources and capabilities to achieve a goal that would otherwise be unattainable.

RELATE focuses the designer on six key concepts of MAS simulations: relationships, environment, laws, agents, things (objects), and effectors. A library of Java classes was developed that enabled the researcher to rapidly prototype an agent-based simulation, supporting cross-platform and web-based designs. Two reference cases were developed that allowed for easy code reuse and modification. Additionally, an existing networked DIS-JAVA-VRML simulation was modified to demonstrate the ability to utilize the RELATE library to quickly incorporate agents into existing applications.

B. COMPOSITE AGENTS

Multi-agent system simulations typically consist of numerous high-level agents that represent entities operating in a common, shared environment. The agents residing in this "outer environment" interact with one another and the objects in the environment. They sense their environment, interpret the sensory input and make decisions as to what actions to take. These actions in turn affect the environment either directly through agent-to-environment interactions or indirectly through agent-to-agent interaction. In an effort to capture the strengths of both cognitive and reactive agents, while at the same time simplifying the design of such a complex agent, a *Composite Agent* architecture has been developed.

Composite Agents are composed of combinations of cognitive and reactive agents (Figure 1). They contain a set of cognitive *Symbolic Constructor Agents* (SCAs) that work with sensory streams (or *impressions*) from the outer environment to create a symbolic inner environment (E_{inner}) representing the agent's perspective of the outer environment (E_{outer}). The SCAs define the agent's sensor capabilities and are tailored to sense specific aspects of the environment. They also act to control and filter *impressions* of the outer environment, so the agent isn't overwhelmed in a rich outer environment. E_{inner} is influenced not only by what the SCAs sense, but also by the CA's internal state. For instance, in a predator-prey simulation, if the predator is hungry and senses an animal, it would show up in E_{inner} as food. On the other hand, if the predator has just eaten, then the animal would appear as just another animal in E_{inner}.

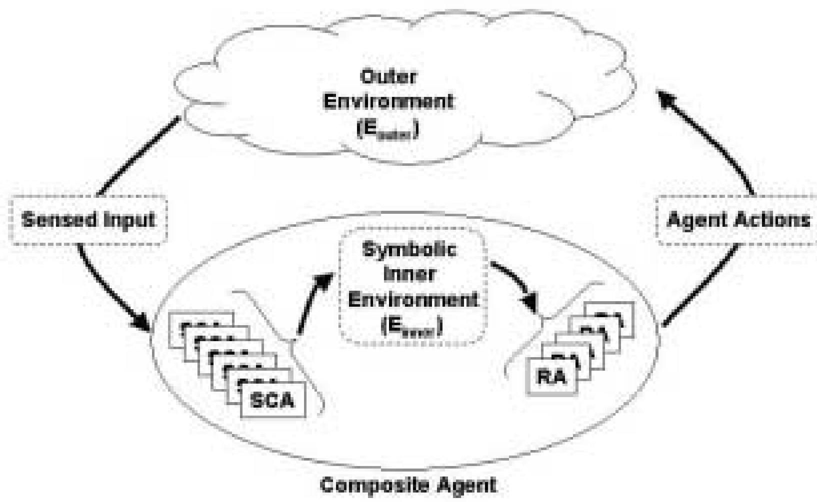

Figure 1 Composite Agent

The symbolic inner environment is the agent's perception of the shared outer environment within which it operates. E_{inner} has little resemblance to the actual outer environment, rather it is an encoding of E_{outer} optimized to suit the Composite Agent's specific function. The role of an SCA is not unlike the role of radio navigation aid used by a pilot. The navigation aid senses radio signals in the outer environment and converts them into directional information that the pilot can use to navigate the aircraft. The inner environment used by the pilot for making decisions has little resemblance to the view looking out the window, but it is optimized for use by the pilot in navigating the aircraft.

Combined with the SCAs is a set of *Reactive Agents* that operate on the symbolic inner environment and generate actions for the CA to perform. Each RA has a set of possible goals and an apparatus for managing the process of selecting the active goal or goals.

C. REACTIVE AGENTS AND GOAL MANAGEMENT

Composite Agents contain numerous *Reactive Agents* (RAs), where each reactive agent is responsible for promoting a specific behavior of the *Composite Agent*. The set of RAs taken as a group, define the *Composite Agent's* set of high-level behaviors. The RAs operate within the world of the inner environment. They take as input sensory information from E_{inner}, and produce as output actions for the agent to perform.

Each RA has one or more goals specific to furthering the RA's behavior or function. So at any given time there are numerous goals competing for the *Composite Agent's* attention. Just as humans have multiple goals (sometimes conflicting), an agent too has multiple goals it wishes to satisfy. In human decision-making, goals are constantly shifting in priority, based on the person's context and state. Agents can mimic the flexibility and substitution skills of human decision-making through the use of a variable goal management apparatus within the RAs. It is from this goal apparatus where contextually appropriate, intelligent behavior emerges. RAs interpret the symbolic inner environment and through their goal apparatus, process this information to balance their goals and return an appropriate action for attaining their highest priority goal or goals (Figure 2).

Goals have four components; a state, a measurement method, a weight, and action or set of actions for achieving the goal. The goal's state is an indication of whether a goal is in an active, inactive, or some other domain specific state. The measurement method translates the sensory input received by the RA into a quantifiable measure of the current strength of a goal and how well it is being satisfied. This permits an agent to prioritize goals and adjust goal states based on context. A goal may also have a weight attached that can be used to adjust the importance or priority of the goal based on experience. Tied to each goal is an action or set of actions for achieving the goals under varying circumstances. The end result is that within the RA goal apparatus there are multiple goals that are constantly changing -- moving up and down -- with the top (active) goals dominating the agent and its behavior.

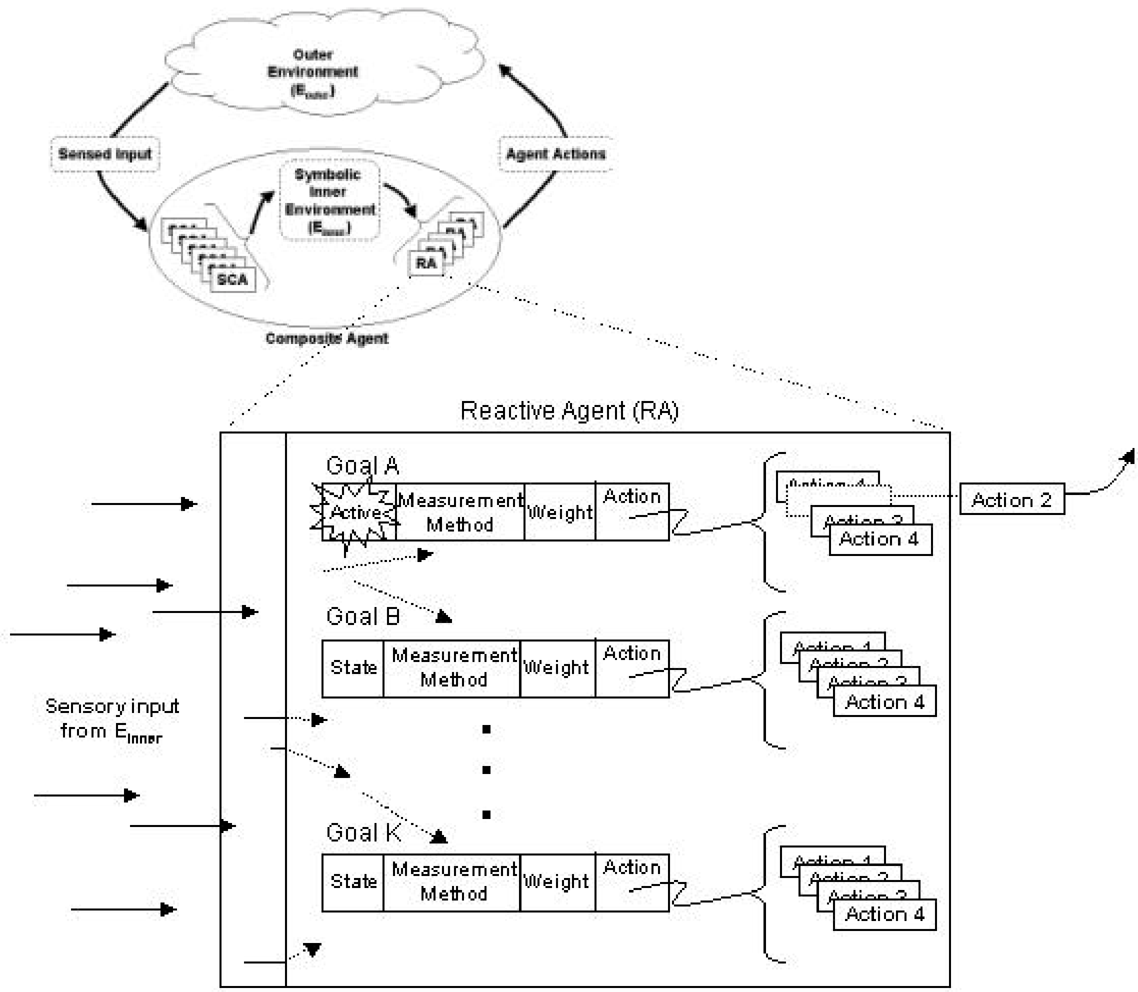

Figure 2 Reactive Agent

Additionally, agents can discard behaviors that do not further their goals, and increase the use of behaviors that have proved successful in reaching goals. This simple behavior serves as a reactive learning system where the agent learns from the environment, based on "what works" with no human expertise or intervention.

Goal switching based on a dynamically changing environment produces innovative and adaptive behavior, however, it is desirable to balance this with doctrinally correct and appropriate actions. This balance is achieved through the encoding of procedural knowledge in a data structure called *tickets*.

D. TICKETS

Symbolic Constructor Agents and the goal apparatus were developed to control the agent's sensory capability and decision-making. In order to provide agents with a rich procedural knowledge base while still supporting adaptive behavior, a data structure called *tickets* has been developed. Tickets allow reactive agents to apply procedural knowledge in context. They define the agent's action set, i.e., its means to achieve its goals. They are used to organize procedural knowledge and provide the ability to balance doctrinal behavior with adaptive, innovative action, resulting in enriched problem solving behavior.

Tied to each of an agent's goals are one or more tickets that define how to achieve the goals. The tickets may have prerequisites or co-requisites that must be met in order for a ticket to be active (see *connectors* below). Additionally, tickets are composed of one or more frames, with each frame being one or more actions or behaviors. Various types of tickets have been defined, with choices ranging from uninterruptible to interruptible, and sequential to non-sequential.

Simply encoding procedural knowledge and linking it to various goals is not sufficient for creating intelligent behavior. The desire is to apply the most appropriate procedures for a given situation. The problem is that in a dynamic system the "given situation" not only changes constantly, but also is so complex, the system designer can't conceive of and account for every possibility. Therefore, the mechanism for determining the "most appropriate" procedures must be flexible and able to support the same level of complexity as the changing contexts of the dynamic system. The ability to take the correct action to match the situation is provided through the use of an apparatus called *connectors*.

E. CONNECTORS

Connectors represent work that is based on symbolic types. They permit logical substitutions and sequencing, and facilitate explanations of reasoning. Connectors are a way to associate impressions, ideas and actions with a given context and achieve a logical sequence of behavior. Connectors are active objects that sense and react to the environment. They activate (extend) and deactivate (retract) based on the current context. As the agent's state and the state of the environment changes, the connectors sense the changes and extend or retract accordingly. By attaching connectors to various elements within the system, including tickets, the connectors signal the elements state of readiness and level of fitness for the current situation. With the connectors continually reacting to the environment, behavioral and procedural knowledge (tickets) can bind at runtime to fit the context as it develops. This binding is based not only on the state of the environment, but also on the goals of the agent and its social interactions with other agents. In this way, the correct procedural knowledge can be brought to bear in the correct situation.

IV. MOVES AGENT RESEARCH: WHERE WE'VE BEEN

The MOVES Institute has traditionally focused on military related simulations and applications. This focus has led the Computer-Generated Autonomy Group to research projects in the areas of modeling human and organizational behavior, land navigation, tactical land combat, and integrating agents into networked virtual environments. More recently the group's research has spread into the areas of interactive story generation, computer security modeling and simulation, and MAS simulation auto-narration.

A. LAND NAVIGATION AND TACTICAL LAND COMBAT

One of the initial projects undertaken by the Computer-Generated Autonomy Group created a tactical helicopter reconnaissance model to support planning for the testing and evaluation of the Comanche helicopter acquisition cycle. The model served as a simulation laboratory for scenario planning, requirements forecasting, and platform comparison analyses [Unrath, 2000]. The model integrated adaptive tactical navigation with agent sensory and weaponry system characteristics. Agents determined their movement direction based on their perceived environment and movement personalities (Figure 3). It incorporated a three dimensional aspect to properly simulate aerial reconnaissance and an integrated graphical user interface (GUI) that allowed users to develop environments, instantiate agent propensities and attributes, and set simulation parameters. It captured simulation summary statistics that illustrate enemy performance, helicopter performance, and logistical requirements. The resulting model demonstrated the ability to represent helicopter reconnaissance behavior and established an initial simulation tool to further explore Comanche operational planning.

A second project in the area of navigation focused on developing a detailed cognitive model of expert tactical land navigation. Tactical land navigation is an extremely important, but difficult task, performed daily by small unit leaders. Interviews with experts at the U. S. Army Special Forces Qualification Course formed the basis of a cognitive model. A multi-agent system was developed to computationally represent the route-planning portion of the performance model [Stine, 2000]. This model demonstrated that a MAS could accurately and realistically represent human performance modeling in a simulation.

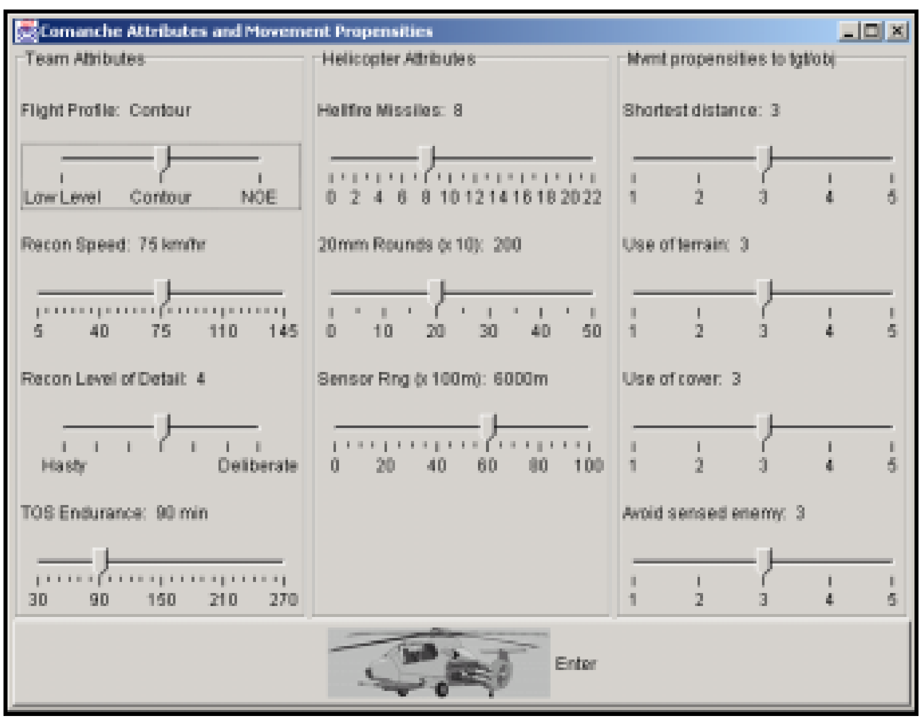

Figure 3 Comanche helicopter agent attributes and movement propensities

B. MODELING HUMAN AND ORGANIZATIONAL BEHAVIOR

Over the past 60 years, the U.S. Army has undergone numerous reorganizations, and each time, a major testing program is involved. However, these testing programs normally focus on the strategic level and less attention is paid to the effect of the proposed changes at the tactical level. Unit leaders are forced to incorporate new equipment or reorganize without an understanding of what the real effects will be (at the unit level) of the change.

The military modeling and simulation community has attempted to address this issue but the current set of single entity simulations are limited in their ability to replicate dynamic complex behavior. A MAS simulation was created (GIAgent) which allowed analysts to gain an understanding of the effects of changing the organization of a company-level infantry unit, as well as experiment with the complex relationships between maneuver and unit organization without putting the unit in the field. Figure 4 depicts the interior of an agent including the agent's personality attributes, movement goals, combat goals and sensed environment. Also shown is the agent's commitment to achieving its goals, whether that is to attack, defend, perform reconnaissance or ensure its own survival. This screen, known as the "Brian Lid," provides a snapshot of the agent and allows the developer to determine what the agent is doing and why. The GIAgent software was a second-generation model built on the RELATE architecture.

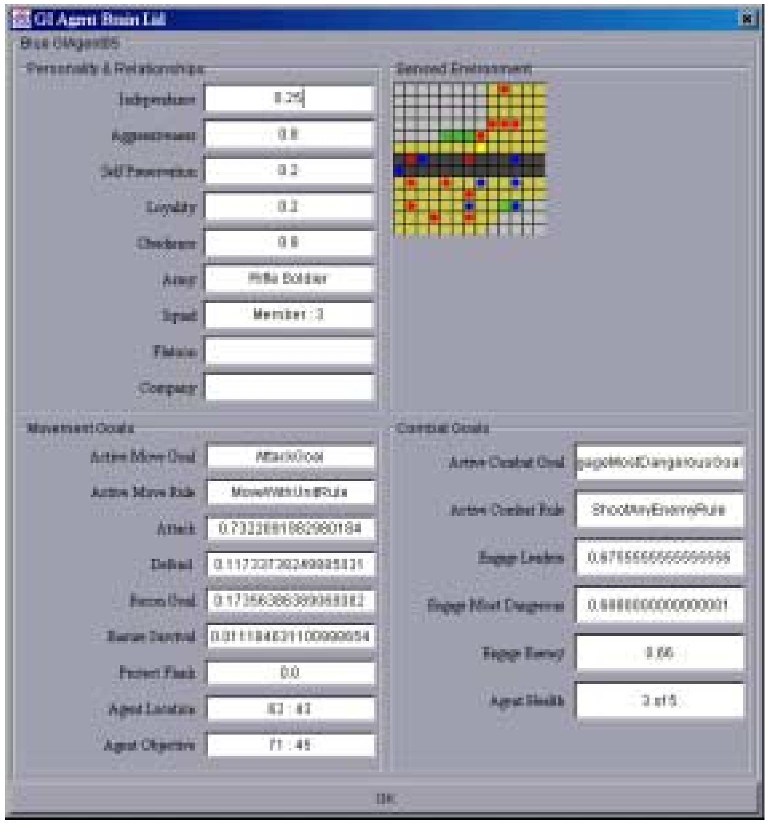

Figure 4 Agent interior from GIAgent

Follow-on research explored the inherent similarities between the numerous ground combat entities and ground combat operations [Mert and Jilson, 2001]. Careful analysis of the generalized concepts in combat entities and combat operations provided a framework to assist developers in modeling many ground combat situations with a single simulation. This research used three distinct MAS combat models to illustrate the generalization framework, including a model created in-house called GENAgent, which was a third generation MAS based on a redesign of GIAgent.

As these MAS simulations advance the level of realism in modeling soldiers operating on a battlefield, accurately capturing the agents perception of the environment becomes increasingly important. The Composite Agent (CA) architecture described earlier models an agent's perception in two phases. First is the processing of incoming sensory data (by the SCA) from the environment (E_{outer}), and second is the translation and storage of the data in the agent's inner environment (E_{inner}). The CA architecture, and two-phased perception model, lends itself well to introducing variability and non-homogeneity into different agents. By controlling the level of hindrance or interference the agent realizes when constructing its inner environment, individual differences in information processing can be modeled, as well as environmental factors impacting sensory data.

A study involving a simple path-finding task was undertaken to determine the overall utility of this architecture with respect to truly representing human performance in cognitive tasks [Hennings, 2001]. Humans as well as agents were put through the same tasks in their respective environments. While some agent combinations were statistically the same as human behavior, a more important finding emerged indicating that agents capable of adapting to their environment and using different path-finding techniques could be created with the Composite Agent architecture.

C. AUTONOMOUS AGENTS AND NETWORKED VIRTUAL ENVIRONMENTS

Networked virtual environments are valuable tools for many tasks. The graphical representation of environments allows users to visualize the problem space with which they are interacting. They are extremely useful for applications including design, training, experimentation, testing and entertainment. Unfortunately, traditional networked simulations are technologically frozen the moment they are completed. They require prior knowledge of all entities that will be used in the system, along with their graphical representations, implemented behaviors, and protocols. Adding new features requires shutting down, coding, integrating, testing, and recompiling the entire system. NPSNET-V, currently under development at the MOVES Institute, is a novel architecture for networked simulations that supports scalable virtual worlds with built-in dynamic entity loading[http://movesinstitute.org/~npsnet/v]. That is, new and previously unknown entity types can be added to the simulation without the need for shutting down the system.

By combining the NPSNET-V architecture with a system for creating autonomous, adaptable agents, it is possible to develop virtual worlds supporting a large number of dynamic, heterogeneous entities with complex, adaptable, and interactive behaviors. RELATE (an agent architecture discussed earlier) has been integrated into NPSNET-V to create such a capability. A test-bed application called *FishWorld* was created (Figure 5) which resulted in a networked virtual environment hosting a multi-agent simulation of the Monterey Bay Aquarium Kelp Forest exhibit [Washington, 2001].

FishWorld is a fully dynamic, scalable, networked application that creates a realistic, virtual underwater environment. It is a combination of this virtual environment with an interactive multi-agent simulation architecture that supports a large number of dynamic, heterogeneous entities with complex, adaptable, and interactive behaviors. *FishWorld* is the backdrop for interaction between a myriad of autonomous and user-controlled agents of varying types, each with unique personalities. It is highly scalable, and able to host a large number of heterogeneous agents. The agents are able to interact with the environment, be affected by currents, and be affected by environmental collisions. The heterogeneous, autonomous agents that populate *FishWorld* not only interact with each other, but also new fish types that are added dynamically. The agents learn about and adapt to the new additions without using deterministic algorithms or scripted behaviors. The determination of which agent will be dominant or be most successful is left up to the agent that best adapts. If there are many different types of predators, the food chain is determined by natural selection. Rigid, non-adaptable agents may emerge dominant in the short term, but may in turn become dominated by agents that are adaptable.

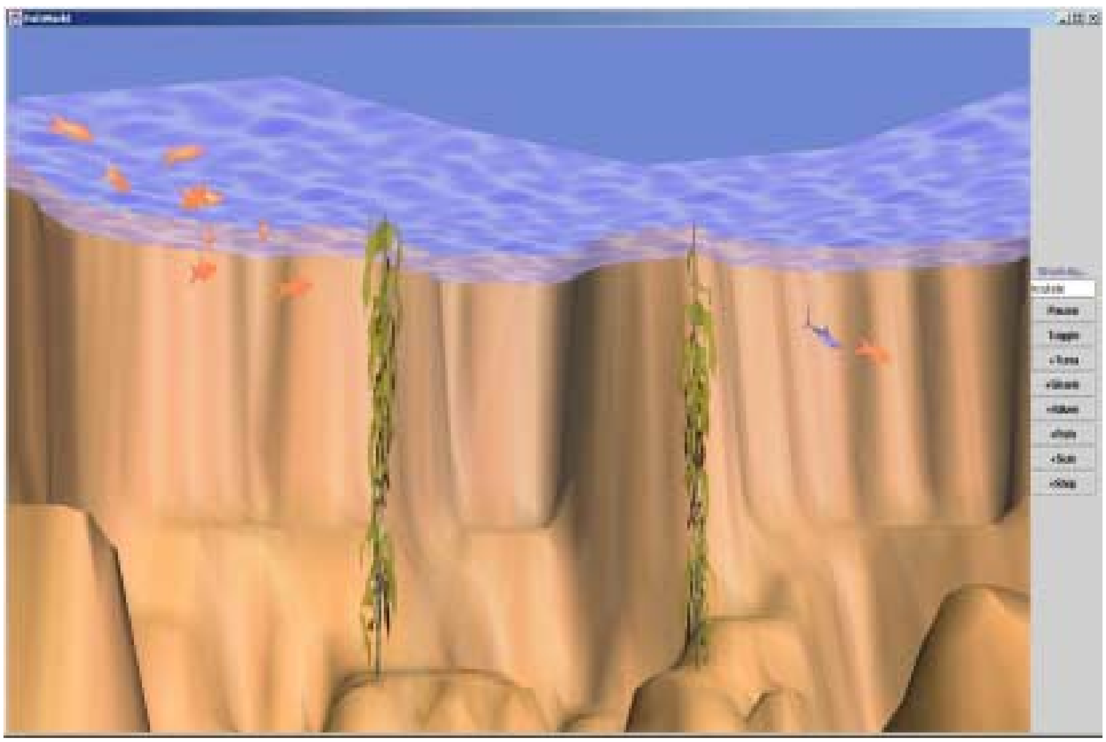

Figure 5 FishWorld

The ability to combine an agent architecture, like RELATE, with NPSNET-V makes possible the creation of test-bed applications useful for experimentation –

integrating improvements from previous iterations for subsequent trials. This process of iteratively testing new subjects in an environment is the process for many industrial, scientific, and military experiments. This is especially useful for the simulation of human participants in automated forces.

V. MOVES AGENT RESEARCH: WHAT'S AHEAD

The multi-generational MAS research and insight gained over the past three years has manifested itself in increasingly complex simulations that were progressively easier to design and implement. This progress has allowed the Computer-Generated Autonomy Group to branch off into some very diverse areas of research. These research projects represent exciting new directions for the MOVES Institute. The domains include interactive story generation, modeling of the computer security domain, and agent-based simulation auto-narration.

A. COMPUTER GENERATED INTERACTIVE STORIES

The Department of Defense (DoD) uses modeling and simulation for a variety of purposes, such as to conduct joint training exercises, develop and evaluate new doctrine and tactics, analyze alternative force structures, and study the effectiveness of new weapons systems. Advances in information technology have lowered the cost of computer-based models and simulation, making modeling and simulation a cost-effective alternative to live training and exercises. While these advances have gone a long way towards creating technically accurate simulations they have not addressed the issue of presenting realistic scenarios while supporting user interaction.

The goal of interactive simulation, whether it is a virtual story or a combat simulation, is to present the user with an experience that suspends their disbelief in the artificialities imposed by the system. In this way, the user feels it is a "real" experience. From the DoD perspective, this results in more realistic and effective training, as well as more accurate assessments of the systems, tactics or doctrine being evaluated.

The entertainment industry has long known that to achieve this suspension of disbelief, it is not sufficient to simply produce a technically accurate simulation. It is the unfolding of events and presentation of the story, along with rich believable characters that makes for a truly effective and immersive experience. The Computer-Generated Autonomy Group is exploring the use of autonomous agent technology to guide the behavior of the simulation characters, while constructing a dynamic, interactive story line that is free to unfold based on the actions of the user, the internal states of the autonomous characters, the laws of the simulation world and the global state of the simulation environment.

A system capable of controlling the actions of autonomous computer generated characters within the guidelines of a story or simulation scenario must support complicated worlds with multiple characters and rich plot complications. At the same time, it must be adaptable to multiple domains, whether it be presenting training scenarios in a ground combat simulation or immersing the user in an action-adventure story.

Current approaches based on artificial intelligence planning techniques can support complicated plots with a diverse set of story characters, but they are extremely domain-knowledge specific. Extensive time and effort is required to generate new knowledge bases and dependency networks for each new story. Algorithmic approaches using tree or graph structures to store story events provide a domain independent methodology, but for complicated stories, the tractability of these knowledge structures can be overcome by the combinatorial problem of evaluating all possible plots each time an event occurs [Weyhrauch, 1997]. The problem of creating a general interactive story system is one of developing an architecture that scales well and is domain independent.

The Computer-Generated Autonomy Group has developed an interactive, agent-based story system based strongly on the use of *tickets* and *connectors* to present highly interactive and dynamic stories. A typical story consists of goal driven autonomous characters, a narrative structure aligned closely with the protagonist, and a collection of potential scenes, along with media, dialog and character interactions to populate the scenes. These story elements are combined dynamically at runtime to generate a story that adapts to the participants interaction and the state of the participant's character.

Figure 6 is a screenshot of a scene in which two autonomous characters are conversing in front of a building. The selection of the specific scene within the context of the story is non-scripted. A stage manager agent selects the scene to be played based on many different criteria. Some of these include the protagonist's personality, what the protagonist has experienced thus far in the story, and where the story is with regards to its progression through its narrative phases. Likewise, the interactions between the two characters as the scene plays out, and the consequences of those interactions, are non-scripted. The story is in essence self-organizing, built from the bottom up from a pool of story elements. By taking a bottom up approach, the system is able to overcome the scaling and complexity problems of traditional AI based methods while supporting domain independent story content.

Figure 6 Two autonomous characters conversing

B. COMPUTER SECURITY

The information security domain is a dynamic and vastly complex environment, and security researchers lack the tools required to analyze and understand this extremely complex environment. The field currently does not have any widely accepted "information physics", nor does it have a complete model of the domain that includes the human aspect of the problem. A simulation is being constructed that can not only answer researcher questions, but also can provide insight into the direction the field is moving. The MOVES Institute is developing a virtual laboratory to simulate the environment of information security, creating a virtual battle space for network security research. While the system can investigate specific hypotheses, the true power of the system will be to provide inductive insight. By allowing the system to evolve as a complex adaptive system, computer security researchers may gain insight into the evolutionary patterns of the domain. The problem domain is being modeled with a coevolving landscape, whereby the battlefield (information components and architectures) and war fighters (users, administrators, attackers, defenders, etc.) evolve continuously throughout the simulation. This coevolving landscape permits the information systems to upgrade as new hardware is deployed and software installed. It permits the actors to learn skills, develop relationships, change goals and behaviors, and develop new tools, tactics, and procedures as the environment evolves. The effects of adding and removing users can be examined, as well as the effects of dynamically reconfiguring networks, constantly upgrading hardware and software, and varying the level of training provided to the users.

C. AGENT-BASED SIMULATION AUTO-NARRATION

One of the most exciting research projects currently underway is an agent based simulation auto-narrator. When watching MAS simulation demonstrations with dots moving about a screen, a human narrator describes what the dots are doing. But is this interpretation and narration of the agent actions coming from the narrator or from the model? Until the models narrate their own behavior there is no way to know. Through the use of self-documenting *connectors*, analysts will not only be able to study behavior in terms of “what” happened, but the models themselves will provide insight as to “why” it happened.

VI. CONCLUSION

When a totally new research direction was added to the MOVES Institute's charter, the goals were set high, with expectations to match. It was well understood that multi-agent systems (MAS) simulation and autonomous behavior have tremendous potential for application in defense and entertainment/defense projects. It was also understood that building a strong research group takes time. However, the Computer Generated Autonomy Group has made tremendous progress in bringing MAS simulation techniques to Department of Defense (DoD) models and simulations, and advancing the start-of-the-art to make adaptive behavior far easier to create and control. Research projects in route planning, land combat, cognitive modeling of land navigation, and modeling organizational changes in military units have provided valuable insight into their respective problem domains and been well received by their DoD sponsors.

But this work is just the beginning. The technologies introduced here place us on the forefront of some exciting new applications and projects. In the not too distant future, the methodology and tools for creating MAS simulations will be as accessible as those currently available for traditional discrete-event simulations. These breakthroughs, coupled with the Institute's experience with networked virtual environments promises to produce some exciting virtual worlds for training, experimentation and simulation.

LIST OF REFERENCES

[Arthur, 1994] Arthur, B. 1994. Inductive Reasoning and Bounded Rationality. *American Economic Association Papers*, 84: 406-411.

[Ercetin, 2001] Ercetin, A., *Operation-Level naval Planning Using Agent-Based Simulation*. Master's thesis, Naval Postgraduate School, Monterey, CA, 2001.

[Ferber, 1999] Ferber, J., *Multi-Agent Systems, An Introduction to Distributed Artificial Intelligence*. Addison-Wesley, Harlow, England, 1999.

[French], 2001] French, S., *Analyzing Projected Personnel Retention Utilizing Complex Adaptive Systems,* Master's thesis, Naval Postgraduate School, Monterey, CA, 2001.

[Hennings, 2001] Hennings, C., *Designing Realistic Human Behavior into Multi-Agent Systems* Master's thesis, Naval Postgraduate School, Monterey, CA, 2001.

[Hiles, 1999] Hiles, J., *Course Notes for MV-4015 Agent-Based Autonomous Behavior for Simulations*. Winter, 2000, Naval Postgraduate School, 1999.

[Ilachinski, 1997] Ilachinski, A., Center for Naval Analyses, *Irreducible Semi-Autonomous Adaptive Combat (ISAAC): An Artificial-Life Approach to Land Warfare*, CRM 97-61.10/August 1997.

[Mert and Jilson, 2001] Mert, E. and Jilson, E., *Modeling Conventional Land Combat in a Multi-Agent System Using Generalization of the Different Combat Entities and Combat Operations***,** Master's thesis, Naval Postgraduate School, Monterey, CA, 2001.

[NRC, 1997] Zyda, M. and Sheehan, J. (eds.). *Modeling and Simulation: Linking Entertainment & Defense*, National Academy Press, September 1997.

[Pawloski, 2001] Pawloski, J., *Modeling Tactical Land Combat Using a Multi-Agent System Design Paradigm (GI Agent)*, Master's thesis, Naval Postgraduate School, Monterey, CA, 2001.

[Reynolds, 1987] Reynolds, C. (1987) "Flocks, Herds, and Schools: A Distributed Behavioral Model", in *Computer Graphics*, 21(4) (SIGGRAPH 87 Conference Proceedings) pages 25-34. http://www.red3d.com/cwr/papers/1987/boids.html, 30 July 00.

[Roddy and Dickson, 2000] Roddy, K. and Dickson, M., *Modeling Human And Organizational Behavior Using A Relation-Centric Multi-Agent System Design Paradigm*, Master's thesis, Naval Postgraduate School, Monterey, CA, 2000.

[Stine, 2000] Stine, J., *Representing Tactical Land Navigation Expertise*, Master's thesis, Naval Postgraduate School, Monterey, CA, 2000.

[Unrath, 2000] Unrath, C., *Dynamic Exploration of Helicopter Reconnaissance Through Agent-based Modeling*, Master's thesis, Naval Postgraduate School, Monterey, CA, 2000.

[Washington, 2001] Washington, D., *Implementation of a Multi-Agent Simulation for the NPSNET-V Virtual Environment Research Project*. Master's thesis, Naval Postgraduate School, Monterey, CA, 2001.

[Weyhrauch, 1997] Weyhrauch, P., *Guiding Interactive Drama*. Ph.D. thesis, Technical Report CMU-CS-97-109, School of Computer Science, Carnegie Mellon University, 1997.

[Weiss, 1999] Weiss, G., Multiagent Systems, *A Modern Approach to Distributed Artificial Intelligence*. MIT Press, Cambridge, Massachusetts, 1999.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center 8725 John J. Kingman Rd., STE 0944 Ft. Belvoir, VA 22060-6218	2
2.	Dudley Knox Library, Code 013 Naval Postgraduate School Monterey, CA 93943-5100	2
3.	Research Office, Code 09 Naval Postgraduate School Monterey, CA 93943-5138	1
4.	Director, MOVES Institute Naval Postgraduate School Monterey, CA 93943-5138	4
5.	Professor John Hiles MOVES Institute Naval Postgraduate School Monterey, CA 93943-5138	2
6.	Major Michael VanPutte MOVES Institute Naval Postgraduate School Monterey, CA 93943-5138	2
7.	Commander Brian Osborn MOVES Institute Naval Postgraduate School Monterey, CA 93943-5138	2