Simulation & Gaming
http://sag.sagepub.com/

Integrating crowd-behavior modeling into military simulation using game technology

Frederic D. McKenzie, Mikel D. Petty, Paul A. Kruszewski, Ryland C. Gaskins,
Quynh-Anh H. Nguyen, Jennifer Seevinck and Eric W. Weisel
Simulation Gaming 2008 39: 10 originally published online 17 December 2007
DOI: 10.1177/1046878107308092

The online version of this article can be found at:
http://sag.sagepub.com/content/39/1/10

Published by:
SAGE
http://www.sagepublications.com

On behalf of:
Association for Business Simulation & Experiential Learning
International Simulation & Gaming Association
Japan Association of Simulation & Gaming
North American Simulation & Gaming Association
Society for Intercultural Education, Training, & Research

Integrating crowd-behavior modeling into military simulation using game technology

Frederic D. McKenzie
Old Dominion University, USA
Mikel D. Petty
University of Alabama in Huntsville, USA
Paul A. Kruszewski
GRIP Entertainment, Canada
Ryland C. Gaskins
Old Dominion University, USA
Quynh-Anh H. Nguyen
Old Dominion University, USA
Jennifer Seevinck
Old Dominion University, USA
Eric W. Weisel
WernerAnderson, USA

Crowds of noncombatants play a large and increasingly recognized role in modern military operations and often create substantial difficulties for the combatant forces involved. However, realistic models of crowds are essentially absent from current military simulations. To address this problem, the authors are developing a crowd simulation capable of generating crowds of noncombatant civilians that exhibit a variety of realistic individual and group behaviors at differing levels of fidelity. The crowd simulation is interoperable with existing military simulations using a standard, distributed simulation architecture. Commercial game technology is used in the crowd simulation to model both urban terrain and the physical behaviors of the human characters that make up the crowd. The objective of this article is to present the design and development process of a simulation that integrates commercially available game technology with current military simulations to generate realistic and believable crowd behavior.

KEYWORDS: *AI; commercial game technology; Crowd Federate; crowd modeling capability; crowd models; crowd simulation; design; development; distributed simulation; fidelity; game AI; human behavior models; integration; military simulation; noncombatant crowds; realism; tactical training applications; simulation architecture*

Crowds of noncombatants play a large and increasingly recognized role in modern military operations and often create substantial difficulties for the combatant

AUTHORS' NOTE: The research described in this article is sponsored by the Defense Modeling and Simulation Office, the Air Force Research Laboratory, and the Joint Forces Command under Contract F33615-03-C-6340. That support is gratefully acknowledged.

SIMULATION & GAMING, Vol. 39 No. 1, March 2008 10-38
DOI: 10.1177/1046878107308092

forces involved. U.S. military actions in Mogadishu, Bosnia, Afghanistan, and Iraq exemplify the significant effects that crowds may have on military operations. However, in spite of their potential significance, realistic models of crowds are essentially absent from current military simulations. For the scenarios considered likely in future conflicts, the absence of crowds and of noncombatants in general would be a serious departure from realism.

The Virginia Modeling, Analysis and Simulation Center (VMASC) is engaged in a multiphase research project aimed at developing a crowd-modeling capability for military simulation. The first phase, now complete, consisted of three parts: a requirements analysis to identify military simulation crowd-modeling requirements, a literature survey to examine psychological research relevant to crowd modeling, and a design study to explore design issues in the implementation of a crowd simulation (Petty, McKenzie, & Gaskins, 2003; Petty, McKenzie, Gaskins, & Weisel, 2004). The findings of the first phase included two discoveries that served to focus development efforts in the second phase. First, we found that the greatest requirements for a crowd-simulation capability were in real-time tactical training applications. Therefore, our intention is to use the crowd simulation for that application. Second, we realized that there is an important distinction between crowds (hundreds to thousands of people) and populations (tens of thousand to millions of people) in terms of size, behaviors, duration, extent, and effects on military operations. As detailed below, the second phase is focused on simulating crowds rather than populations. In the second phase, we are developing a crowd simulation that will be interoperable with existing military simulations and will have a credible psychological basis for the crowd behavior that it generates.

The sections of this article cover these topics: background on distributed simulation, crowd-simulation requirements, and relevant psychological research; the architecture and development status of the Crowd Federate; the processes used to develop scenarios and terrain for the crowd simulation; and the tools, methods of integrating commercial game military-simulation components. The last topic is presented in some detail.

Background

This section provides background information on military-distributed simulation, the context in which our crowd simulation operates, and describes the preparatory requirements analysis and literature survey conducted before beginning development of the simulation.

Military-distributed simulation

Distributed simulation is a broad term that we refer generally to sets of simulations that have been networked (or otherwise connected) and are collaboratively simulating a common virtual world. A set of flight simulators linked via a local area network into a single battle and massive multiplay games are both examples of distributed simulation. Military-distributed simulations often include multiple heterogeneous simulations, including some with human-in-the-loop components, to accomplish a targeted goal, such as planning, mission rehearsal, or training. The term *distributed interactive simulation* can refer to both the practice of using connected simulations and also to an established protocol standard for integrating those simulations.

The U.S. military has developed a series of distributed simulation protocols and architectures, including the High Level Architecture (HLA), which is used in this project. HLA is an architecture standard for constructing distributed simulations that facilitates interoperability among different simulation systems and types and promotes reuse of simulation software modules (Kuhl, Weatherly, & Dahmann, 1999). Within HLA, a set of collaborating simulations is called a *federation*, a run of the distributed simulation is a *federation execution*, and each of the collaborating simulations is a *federate* (and thus our crowd simulation is referred to hereinafter as the *Crowd Federate*). The HLA concept has been mandated for military simulations and is widely in use today.

Requirements analysis

VMASC consulted with modeling and simulation (M&S) users in the joint community (and others) regarding their current and anticipated needs for crowd modeling in military simulations and surveyed published sources calling for crowd modeling. We used an adaptation of a methodology previously proposed for analyzing requirements in the domain of human-behavior modeling (Chandrasekaran & Josephson, 1999). That method asserts that the intended purposes of a simulation should determine its fidelity requirements.

We added an initial step to the method wherein the simulation purpose was determined from application area[1] and warfare level.[2] A combination of application area and warfare level constitutes a use category; each use category specifies a class of simulation purposes. From those purposes, entity types, cognitive tasks, and fidelity can be determined. However, in our analysis, attention was restricted to crowd entities. Following the revised method, crowd modeling requirements were analyzed in each of 12 use categories, the possible combinations of the four application areas, and the three levels of warfare. The analysis was done by interviewing military simulation experts, focusing on simulation users rather than developers, and surveying military-simulation literature for assertions regarding crowd-modeling requirements. More details about the breakdown of these requirements can be found in McKenzie, Xu, Nguyen, and Petty (2004).

There was a good consensus that crowds were needed in military simulation. An example military requirement was, “[Crowd models are needed for] displaced personnel requiring humanitarian assistance; such persons consume logistical supplies, requiring re-supply of food and water, and affect logistics of combat operations.” However, there was less agreement on what the specific requirements were and even less on how the requirements were expressed. The requirements, both as given by the users and drawn from published sources, were stated in several different ways: needed crowd behaviors (e.g., “take hostile action against combatants”), military mission types for which crowds were generally needed (e.g., “urban warfare”), and effects that a crowd might have on a scenario that needed to be modeled (e.g., "road congestion"). Because of the wide variety of crowd-modeling requirements identified, it seems clear that a single crowd-behavior model is unlikely to satisfy all of the requirements. The largest number of requirements was found at the tactical warfare level and for the training application area for which there were 27 requirements from 12 different sources. So it is within this level and area that we have concentrated our research.

Literature survey

A survey of the current state-of-the-art in crowd modeling was conducted from two perspectives. The first perspective was psychological; here, we surveyed the psychological research literature for research relevant to understanding and modeling the behavior of crowds. Special attention was given to research that considered crowd behavior in military scenarios, but other scenarios, including civil unrest and sporting event riots, were also considered. More than 50 sources for both descriptive and predictive models associated with crowd modeling were studied. The literature surveyed drew largely from the cognitive psychology, social psychology, sport psychology, sociology, police, and military literature. Research of primary interest was noncombatant crowd behavior during military operations.

The psychological literature had many sources that describe and categorize crowd behavior in a qualitative way. Four main crowd types have been identified, with differing behavioral tendencies: aggressive, escapist, acquisitive, and expressive (Varwell, 1978). A crowd can exist for any combination of these reasons or change its type because of the unfolding situation (Moore, 1990). During the time they exist, crowds seem to pass through three specific stages of behavior: assembling, gathering (perhaps "rallying" would be more descriptive), and dispersal (Kenny et al., 2001). During the assembly stage, a planned or environmental event provides the process and motivation behind the movement of people toward a somewhat co-located area. During the rallying stage, a crowd begins to form and groups within the crowd start to engage in group behaviors. These collective behaviors may be peaceful actions, such as singing or chanting, or possibly evolve into violent behaviors involving weapons. Eventually, this crowd process will discontinue its collective rallying and transition to its third and final phase of dispersing, which may be either forced or routine (Bliss, Fallon, Headen, & Gaskins, 2004). A large number of situational, cultural, and personal factors have been identified and documented in the literature as affecting crowd behavior in each stage.

The second perspective was engineering; here, we identified models and simulations with capabilities relevant to crowd modeling that have been or are being implemented as computer systems. Both computational models of crowd cognitive behavior and crowd physical behavior were of interest. VMASC assessed the capabilities of those systems. Real-time applications pose the challenge of handling interactions among crowd agents as well as the limited computational resources available.

Approaches to crowd modeling have been developed from a number of more basic fields of study. Reynolds (1987) first studied distributed behavior modeling for simulating the aggregate motion of a flock of birds. He revolutionized the animation of flocks of animals, in particular birds, or “boids,” by using theories from particle systems relating each individual bird to a particle. A more recent particle-system-based tool is EXODUS (Filippidis, Galea, Gwynne, & Lawrence, 2006), used in fire evacuation studies. Bandini, Vizzari, and Manzoni (2004) modeled crowds using cellular automata to represent and influence navigation within the environment. Others have used biologically inspired insect swarms to model crowd panic responses using, for example, Ant Colony Optimization (ACO; Bonabeau, Dorigo, & Theraulaz, 1999; Banarjee, Grosan, & Abraham, 2005).

Individual and collective actions were studied in what McPhail, Powers, and Tucker (1992) prefer to call *temporary gatherings*. A technique utilizing a combination of particle systems and transition networks to model human crowds in the visualization of human spaces employed by Bouvier and Guilloteau (1996) was the basis of later work in agent dynamics. Brogan and Hodgins (1997) addressed the issue of significant dynamics in simulating group behaviors. Their work simulated groups of creatures traveling in close proximity whose motions are guided by dynamical laws. The focus to this type of animation is on collision avoidance.

Virtual humans have been represented through dynamically generated impostors in the work of Aubel and Thalmann (2000). Tecchia, Loscos, and Chrysanthou (2002) employed image-based methods for real-time rendering of animated crowds in virtual cities. O’Sullivan et al. (2002) presented the Adaptive Level of Detail for Human Animation (ALOHA) model of crowd and group simulation. The ALOHA model incorporates levels of detail for not only geometry and motion but also includes a complexity gradient for natural behavior both conversational and social.

A few models have attempted to examine more general crowd behaviors, integrating several subcomponents, such as collision avoidance, path planning, higher level behaviors, interactions, or giving up (Musse & Thalmann, 2001; Ulicny & Thalmann, 2001). In Musse and Thalmann’s ViCrowd system, a virtual crowd can be created in which the individuals have variable levels of autonomy, including scripted and rule-based, and are guided interactively by the user (Musse, Garat, & Thalmann, 1999; Musse & Thalmann, 2001).

Ulicny and Thalmann (2001) advocate a real-time crowd simulation with an emphasis on individuals in contrast to groups in a multi-agent system. Levels of variety range from zero variety, for which there is only a single solution for a given task, to Level 1, when it is possible to make a choice from a finite number of solutions, and Level 2, in which it is able to use solutions chosen from an infinite number of possible solutions. Seo and colleagues exemplify such a model whereby a Level 1 system presents a crowd that is composed of multiple humans selected from a predefined set composed of sets of exchangeable parts such as heads, bodies, and textures (Seo, Yahia-Cherif, Goto, & Magnenat-Thalmann, 2002). A Level 2 system described in the same study displays a potentially infinite number of unique humans generated by a parameterized anthropometric model generating humans with different morphologies. Ulicny and Thalmann (2002) point out that such higher levels of variety can be unnecessary when perfect visualizations are distracting, and simpler uniform visualizations could help emphasize problems.

Ulicny and Thalmann's (2002) model of the world recognizes the importance of internal psychological or physiological states of agents. These include memory, fear, mobility, or level of injuries in simpler states. Higher-level complex behaviors include wandering, fleeing, or following a path.

Behavior models of Ulicny and Thalmann (2002) focus on the perception of the agents in their surrounding environment. This includes reaction to changes, to other agents, and to the real humans interacting in the virtual world. Behaviors, which can be mixed as needed, range in level from simple scripted behaviors to high-level autonomous behaviors such as wandering, fleeing, neutralizing the threat, or requesting and providing help. Ulicny and Thalmann's model is a good example of efficiently managing variety. The levels of variety for multiple agents are perhaps one of the most advanced in the concept of crowd simulation. Their current work focuses on enhancing the levels of variety and incorporating motion models that are able to synthesize variations of a given motion (Lim & Thalmann, 2002). Further research is called for to enhance the behavior models by including new behaviors based on sociological observations from gatherings in real-world settings.

Psychological studies

We are conducting psychological studies intended to gather information for a psychologically based cognitive model in the Crowd Federate. The output of these studies are intended to provide important parameters for crowd behavior that must be taken into consideration and also stochastic values for the type of behavior that would occur given a particular context. Thus far, the project has taken a three-pronged approach to these studies: (a) naturalistic observation, (b) interviewing subject-matter experts, and (c) survey research.

Naturalistic observation. The Seattle Police Department supplied 5 hours of raw video footage of protests at the World Trade Organization (WTO) meeting from November 29, 1999 through December 3, 1999. The goal of the demonstrators was to disrupt the WTO meeting. To analyze the crowd behavior exhibited at this event, 6 graduate and undergraduate students participated as subjects. Subjects coded the frequency of occurrence of 55 behaviors in increments of 10 minutes. Two subjects went back and rated the level of aggression in increments of 10 minutes. Descriptive statistics were used to analyze the data. Means and percentages were calculated to determine which behaviors occurred most frequently. Reliability measures were also calculated.

Some of the most frequently occurring behaviors include standing on elevated structures, yelling and shouting, raising flags, filming by media, and chanting. Aggressive behaviors rarely occurred but were highly effective in disrupting the event when they did occur. These include throwing glass bottles, fighting, jumping on moving vehicles, looting, and blocking the road by groups lying down across the street. Overall, the subjects' observations of the behaviors were reliable. Reliability increased during the course of the 5 hours of observations. Standard deviations were also examined as evidence of reliability. Higher deviations occurred for the more frequently occurring behaviors.

Future research will use more subjects, with each coding fewer behaviors.

Subject-matter expert interviews. Both structured and unstructured interview techniques are being employed with senior level nonlethal weapons trainers and police officers with extensive crowd experience. These interviews are designed to better understand the cognitive processes and relevant issues associated with crowd-member and control-force interactions. For instance, do crowds perceive nonlethal weapons as a threat? Some of the challenges that have been uncovered include cultural sensitivity, difficulty of negotiations to lack of interpreters, distinguishing gunmen, organized and paramilitary tactics, denying the area, what to do with the crowd afterward, arresting the leader, and the generality of current research. Typical flashpoints include lethal weapons pointed at a soldier and the use of irritant chemicals. Some of the best ways to avoid a crowd riot include the presence of interpreters, a means for communicating with the crowd, and the presence of civil affairs people.

Survey research. A survey was developed and field-tested on 6 soldiers on active duty in Iraq. The survey examines various psychological variables and their relation to crowd violence. The survey includes both open-ended items and Likert-type scale items. Open-ended items get at the types of crowd events, demographics of the crowd, the interaction with the crowd, temperament of crowd, events leading up to the crowd event, purpose of gathering, and presence of weapons. The survey also asks questions regarding best practices and attempts to understand what was most effective in dealing with the crowd and reestablishing peace. A number of items question the soldiers' level of experience with crowds. Finally, the survey examines a number of crowd factors from the literature and the aforementioned WTO study. Respondents are asked to rate each factor on a 5-point, Likert-type scale representing the strength of the relation to the crowd event turning violent.

Some sample factors from preliminary analysis that were rated as highly related include presence of instigators within crowd, presence of weapons within crowd, willingness to take risk, peacekeeper aggression, size of crowd, use of alcohol and drugs by crowd, societal acceptance of violence, commitment to cause, and presence of organized crowd leadership (Air Force Research Library, 2003; Bliss et al., 2004). In addition to identifying the strength of the relationships of variables to crowd incidents, we plan to develop factual case studies of soldiers encountering crowd situations in Iraq.

Crowd Federate

This section introduces the Crowd Federate implementation, focusing on its overall architecture and the capabilities of the current version. Later sections will provide details of the federate's components.

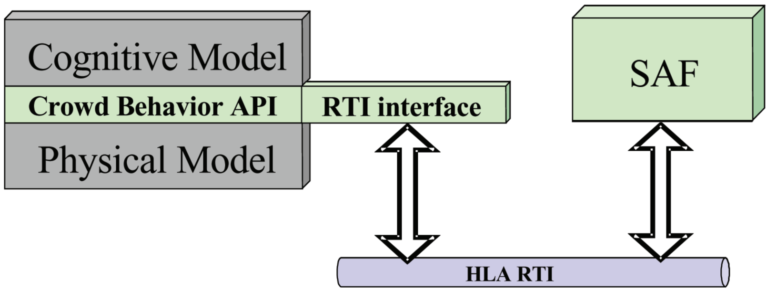

FIGURE 1: Crowd Federate Top-Level Architecture
NOTE: API = applications programming interface; HLA = High Level Architecture; RTI = runtime infrastructure; SAF = Semi-Automated Forces.

Crowd Federate architecture

A central task of the ongoing project is to design, implement, and test a Crowd Federate, for instance, an HLA-compliant simulation that models crowd behavior and is interoperable with a real-time, individual entity-level military simulation. The Crowd Federate will have a multilayered, reconfigurable software architecture, shown at a high level in Figure 1. The architecture separates the model(s) for physical behaviors (e.g., walking, running, route following, stone throwing) from the model(s) for cognitive behavior (e.g., decision making). The cognitive model selects the behaviors that a crowd member will perform, whereas the physical model carries out those behaviors.

The crowd-behavior applications programming interface (API) provides the interface between the cognitive and physical models and is central to the federate architecture. The API allows the integration of separately developed cognitive and physical models. The API is intended to operate in both directions. It facilitates control of the physical model by the cognitive model in providing a repertoire of physical behaviors and drives the cognitive model by relaying physical sensory, event, and state feedback from the physical model to the cognitive model (McKenzie, Xu et al., 2004).

We intend for the architecture to provide a reusable infrastructure within which other crowd models (cognitive or physical) developed by other researchers can be tested. Ideally, separating the cognitive and physical models from each other and linking them with a carefully designed API will make it possible to modify or replace models of one type without affecting the other.

Crowd Federate prototype

Development of the Crowd Federate software is ongoing. The current version of prototype Crowd Federate has working versions of the API, physical model, RTI (runtime infrastructure) interface modules, and the cognitive model framework with limited psychologically based components integrated.

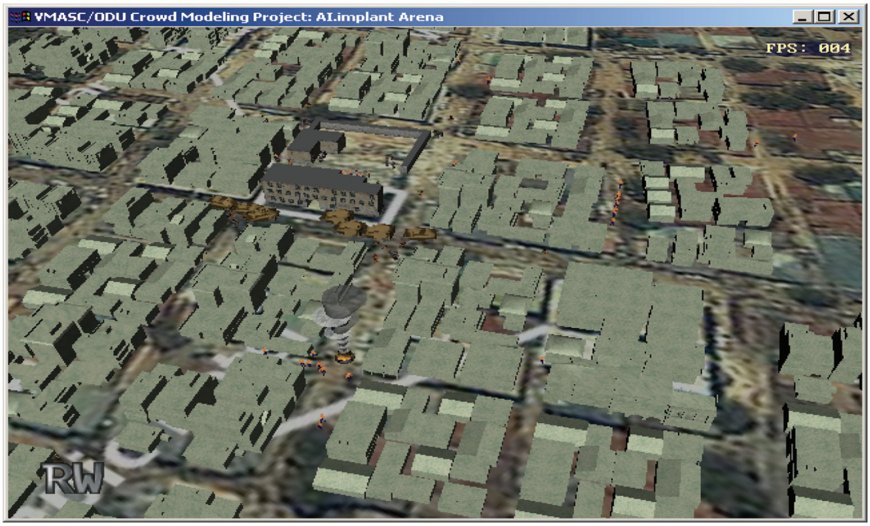

FIGURE 2: Crowd Federate Executing the Mogadishu Scenario

Figure 2 is an image from a test of the Crowd Federate; it shows a moment from the Mogadishu reference scenario (such scenarios will be discussed later). In the figure, the Crowd Federate controls the crowd members, whereas a separate military simulation controls the combatant soldiers and vehicles. Figures 3 and 4 are images of crowds generated by the prototype Crowd Federate. The terrain in these two figures is a digitized version of the urban combat-training facility at the Quantico Marine Corps base.

Cognitive model status

Development of the computational cognitive model is just beginning. As mentioned earlier, the current prototype Crowd Federate has a temporary cognitive model, based on a combination of decision-tree logic and basic crowd-level psychological parameters. As the psychological studies just described start to produce information on which a computational model could be based, we have begun considering alternative modeling paradigms for implementing the cognitive model. This is still an open issue.

Physical model status

The physical model portion of the Crowd Federate is operating well. It is this part of the Crowd Federate that includes the commercial game technology. Details on that integration and the capabilities provided by the game software will follow later.

FIGURE 3: Crowd With Random Movement

FIGURE 4: Crowd With Flocking

Scenario and terrain development

This section describes our notion of reference scenarios, which are detailed recreations in simulation of historical events, and how they are being used for testing and validation. It also details the tools and procedures used to generate terrain databases for the Crowd Federate and the military simulation JSAF (Joint Semi-Automated Forces), using the Mogadishu terrain as an example. JSAF is a constructive simulation of combat that represents military systems at the entity level (e.g., individual vehicles, soldiers, and aircraft are modeled). JSAF provides both physical and behavioral models; it can generate doctrinally appropriate behavior for groups of entities organized into military units. JSAF's behavior generation is mostly autonomous, but it does have an operator interface to allow an operator to direct or override the autonomous behavior. JSAF is extensively used by the U.S. for simulation experiments in the areas of joint doctrine, weapons, and plans.

Reference scenarios

A reference scenario is a highly detailed scenario based on a historical event, with great care taken to reproduce the event as precisely as possible. Included in a reference scenario are: terrain, military force personnel, military force equipment, military mission and rules of engagement, military orders, crowd size and composition, and crowd behavior repertoire. These details, once documented, are implemented as a scenario in the military combat simulation to be used with the Crowd Federate. Currently that military simulation is JSAF.

In our current work, we are developing two reference scenarios. The first consists of two vignettes from the Battle of the Black Sea, Mogadishu Somalia, in 1993. The second involves a nonlethal event in Brcko, Bosnia-Herzegovina. We will only discuss the first of the reference scenarios. In 1993 the United States deployed troops, known as Task Force Ranger, on a peacekeeping mission to Mogadishu, Somalia on the eastern coast of Africa. On the afternoon of October 3, 1993, more than 100 soldiers from Task Force Ranger, supported by 19 aircraft, departed the Task Force Ranger compound on a routine mission to capture two chief lieutenants of Mohamed Farrah Aidid, leader of one of the principal warring factions in Mogadishu. Intelligence placed these men in a building near Bakara Market in Mogadishu, an Aidid stronghold. The plan was to insert Special Forces by helicopter to secure the building. Concurrently, four "chalks," about 12 soldiers, of U.S. Army Rangers would fast-rope from BlackHawk helicopters to defend the perimeter of the target building compound. A ground convoy of 12 Humvees and trucks would then evacuate the troops and prisoners. The mission was planned to take less than 1 hour. In the ensuing action, two BlackHawk helicopters were shot down and a portion of the force was pinned down overnight until rescued the following day. The mission resulted in the capture of two high-ranking Aidid supporters at the cost of 18 U.S. dead and hundreds, perhaps thousands, of Somali militia and civilian casualties. This event is the subject of the well-known book *Black Hawk Down* (Bowden, 1999) and the movie based on it.

The Mogadishu scenario was selected as the subject of a reference scenario based on the following two significant factors:

1. *Well-documented event.* For a reference scenario to effectively represent a historical event that is the subject of simulation, it must be well documented, meaning that a

wealth of reference sources must be available. References should be primary sources when possible. For example, the partial sequence of events found in Rysewyk (1994) is a primary source. In his book *Black Hawk Down,* Bowden states that the Mogadishu event "may well be the most thoroughly documented incident in American military history" (Bowden, 1999, p. 407).

2. *Desired level of crowd interaction.* In the Mogadishu action, the crowd consisted of militia elements and civilian participants, exhibiting behaviors varying from fleeing to obstructing the progress of military forces to combat engagement of military forces. The crowd interaction resulted in a measurable effect on event outcome—specifically, length of operation, military force and crowd casualties, and degrees of mission accomplishment.

Elements of a reference scenario include data concerning terrain, military force personnel, military force equipment, military mission and rules of engagement, military orders, crowd size and composition, and crowd-behavior repertoire. These elements are captured in the execution matrix, sequence of events, and terrain database. The execution matrix details the military mission by breaking out military force personnel and equipment and orders in a matrix format. Each entry is referenced to a source document. The execution matrix represents the intentions of the military force. The sequence of events details the actual events of the historical action and includes crowd as well as military-force events. Like the execution matrix, each entry is referenced to a source document. The terrain database includes imagery, topography, and structure information.

Validation is accomplished by entering the scenario into the combat simulation with and without crowd interaction. Experiments are conducted using varying degrees of crowd interaction. The hypothesis is that the simulation results with a particular level of crowd interaction will be closer to the historical events in the reference scenario than the simulation results at other levels of crowd interaction, including no interaction. Although it is not expected that the simulation would exactly match the outcome of the historical event, similar types of effects can be noted. Such types of effects include mission success, mission delay, amount of delay, fatalities, number of fatalities, equipment destroyed, and so forth.

Essentially, we believe that there exists a level of fidelity of crowd behavior that will most closely match the outcome of the historic event, and that best level is not when the crowd is excluded from the scenario.

SAF scenario development

Developing the Mogadishu scenario in the SAF application, JSAF, proved to be more complicated than expected. The first step was to determine the timeline and to develop the execution matrix in JSAF needed for the military component of the scenario. We found that the types of entities that we needed had not been validated in the version of JSAF that we were using and that many of the behaviors that we needed from those entities did not function correctly. To compensate for those limitations, we had to make compromises and use workarounds to model the scenario.

For example, workarounds were needed in the Mogadishu scenario to model the convoy of Humvees and trucks staged behind the Olympia Hotel, waiting to pick up the prisoners and soldiers. Because there was no predefined aggregate convoy of Humvees and trucks, we had to use individual Humvee and truck entities, placing each one behind the other. We also had to make sure that we manually started each vehicle entity in the correct order, with delays between 5 to 7 seconds to allow them enough time to accept and execute their orders. The workarounds and the need for human intervention made it difficult to recreate a historic event. Nevertheless, we were able to approximate the first vignette of the Mogadishu mission during the prisoner capture operation.

Terrain tools

3D Studio MAX, a common 3D modeling package for the gaming industry, was used as the primary tool for generating the terrain models. After the terrain was developed in a standard format, we converted it into the special formats required. A Compact Terrain Database (CTDB) terrain file was needed for JSAF, and a Maya file was needed for the 3D battlefield-visualization view. CTDB is a standard terrain database format used in JSAF and other related military simulations. The terrain surface is represented as a faceted surface composed of contiguous triangles. Special terrain features, such as buildings and trees, are also represented.

The tools used in its conversion were Terravista with the Database Automatic Re-Use Technology (DART) plug-in tool, and Multigen-Paradigm Creator. The process employed to generate this environment is described in further detail below.

Generating the Mogadishu terrain

The initial step in generating the environment was to locate an appropriate satellite picture of the area from Space Imaging; Figure 5 shows a screen shot of that image.

The image map was then imported into 3D Studio MAX using the provided scale information on the image. A terrain block was created to scale, which covered the area shown in the picture, and textured with the satellite picture. These steps provided a background for generating the 3D buildings used in the terrain. In parallel, we used another satellite picture with a broader overview to correlate with a CTDB that included the area of interest. The original CTDB was used to georeference all of the work that was done in 3D Studio MAX. Terravista with the DART option was used to convert the CTDB into an Open Flight file, which then was imported into 3D Studio MAX. Once the terrain files were imported into 3D Studio MAX, the next step was to construct the buildings around the areas of interest.

For the construction of the buildings in Mogadishu, extensive source data was not available, so data available from the satellite pictures were used as a guide for tracing the footprints of the buildings and creating an "extrusion" of such footprint to create the buildings. Because each of the buildings was created from the satellite

FIGURE 5: Satellite Photograph of Mogadishu

picture, they were already referenced to the coordinate system of the data set. All of the buildings were given a "generic" texture, and the "target" building was given a more detailed look than the rest of the buildings (see Figure 6).

Once the buildings were created, they were prepared to be translated into a CTDB file for use into JSAF. We were able to use 3D Studio MAX to quickly identify faces, which would provide the attributes "footprint" and "roofline" for the CTDB file. After the faces were identified and duplicated as part of each model, then a conversion was made within 3D Studio MAX to translate the files from the MAX file format to Maya format, for use in the stealth 3D viewer, and to the FLT file format for further work on attribute values necessary to reuse the dataset in a CTDB format.

After the translation of the Mogadishu 3D Studio MAX file into the Open Fight format, we continued to define attribute parameters of the database inside Multigen-Paradigm's Creator. Creator was used to give the proper flag attributes to each of the faces previously identified in 3D Studio MAX as a "roofline" and a "footprint." One of the reasons why the extra step was needed to define these attributes in Creator is because 3D Studio MAX was not originally intended to model 3D environments for real-time simulations. However, for the past 3 years, this tool has been evolving to be the tool of choice for real-time game design.

Once these attributes were defined and the database was saved in the Open Flight format, it was then taken into Terravista to prepare it for final conversion to a CTDB file. We used a plug-in module in Terravista called DART to be able to reuse the

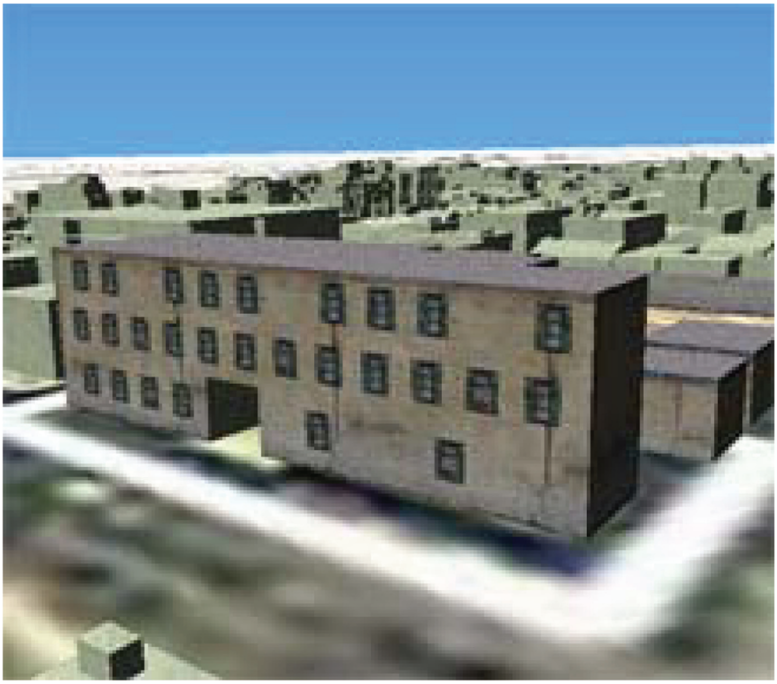

FIGURE 6: Extruded Mogadishu Target Building

original CTDB file and then add the new environment information created with 3D Studio MAX. This procedure involved importing the Open Flight files of all the buildings into Terravista and adding them as culture data.

To do this, we imported an Open Flight vector file, originally created in 3D Studio MAX by using the DART Terrain Converter importer. Once the file was imported, we used the editor tool in Terravista to assign the buildings that we previously imported to the vector file.

Finally the "gaming area" was defined from the terrain tiles, making sure that the gaming area followed CTDB rules for area selection, which in this case was to make sure that an 8-by-8 tile area was used. We then proceeded to build the gaming area with the SAF_USA option enabled, which produces a CTDB file of the built project from Terravista. The resulting CTDB file was used as the terrain database in JSAF (McKenzie, Garcia, et al., 2004).

Integrating game technology

This section discusses the use and integration of commercial game technology into the Crowd Federate. Game technology is applied for two purposes. BGT

AI.implant™ (now Presagis AI.implant) is used for control and composition of the crowd members' low- and mid-level physical behaviors in the Crowd Federate's physical model. BGT Arena™ was used to develop a 3D viewer into our simulated world. This section details how these gaming technologies have been incorporated into our solution.

Character modeling, rigging, and animation

The creation of the simulation character models is constrained by the real-time performance requirements of the end application. Because many figures are being rendered at any one time (a crowd), these must be optimized for maximum realism and performance. Reducing the resolution of textures and minimizing the number of polygons in a figure reduces the computational load on both the rendering engine and the crowd behavioral model.

A lower polygon count is synonymous to a lower resolution and the end application (e.g., the distance of the virtual eye to the crowd) must be considered when determining thresholds, or else the realism of the end product may be compromised. A range of 500 to 800 polygons per figure was determined as optimum for our application.

Modeling and the generation of animation clips that our crowd behaviors can call are done within Maya. Texturing uses open source images and is done using Adobe Photoshop image manipulation software. Each figure model is rigged with an inverse kinematic skeleton. Once this hierarchical system of bones is built to fit the mesh, it is then bound (attached) to the mesh model of the figure and finally configured to enable animation:

1. Skeleton joints are appropriately constrained for animation purposes (e.g., knees don't bend backwards).
2. The relative weighting of bones and their deforming effect on the bound mesh is adjusted for smooth deformations rather than folds in the mesh.
3. Pole vectors and animation handles are defined.

The project required animations of two types: cycling and blending (for between cycles). Cycles such as walking were key framed in Maya, then rendered to clips that the AI.implant behavioral engine can call. When a change in behavior was required, a blend animation clip would be called (e.g., walk to stand), followed by the new animation clip (stand cycle).

Key framing is, however, a fairly laborious and time-consuming process (10 seconds of animation typically takes an animator 40 hours to create). Therefore, motion capture data is now being used. The motions are edited into cycles in Kaydara and the movements are available for standard/common motions from the Kaydara library. Kaydara produces a tool called MotionBuilder that automates many aspects of character rigging and animation. Finally, the export of animation cycles from Kaydara (.fbx file format) creates key frames at every frame, which can then be imported into Maya.

BGT AI.implant for generating behaviors

AI.implant, developed by Bio|Graphic Technologies, is a commercial game AI solution from the entertainment industry that offers a "real-time interactive artificial intelligence animation solution…to create incredibly rich character interactions" (Bio|Graphic Technologies, 2004). Bio|Graphic Technologies (BGT) AI.implant became Presagis AI.implant in August 2007. This software has been used successfully to create digital extras for films and nonplayer characters for video games. The AI.implant world is made up of stationary objects, such as barriers, surfaces, terrains, and paths, as well as two types of dynamic objects, the autonomous and nonautonomous characters, which interact with this world. In the simplest terms, AI.implant controls the behavior of an autonomous character, whereas a nonautonomous character is a character that is not controlled by AI.implant but may interact within the world. The nonautonomous character could be used to model objects such as player-controlled characters, falling rocks, or any other dynamic object such as an SAF that does not have its behavior controlled by AI.implant. The behaviors defined for the autonomous character provide the rules that determine how that character will interact with other objects within the world and generates the steering forces that change the character's position, orientation, and/or speed.

Behaviors. In AI.implant, behaviors are the low-level actions of an autonomous character. Locomotive behaviors are those that affect how an autonomous character moves and may affect a character's direction of motion, speed, and/or orientation. The majority of the behaviors provided by AI.implant are locomotive behaviors that can be further divided into three subgroups, as follows:

1. simple behaviors, which only involve a single character
2. targeted behaviors, which involve a character and a target object
3. group behaviors, which allow characters to act and move as a group, with individuals within the group maintaining about the same speed and orientation.

The available behaviors are listed by type in Table 1.

Many of these behaviors are self-explanatory (e.g., *Orient to, Avoid obstacles,* etc.). However, a few of the targeted behaviors, specifically *Strafe, Go between,* and *Flock with,* may need some clarification. Unlike its dictionary definition, the *Strafe* behavior does not refer to attacking from low-flying aircraft, but instead it is used to cause its character to "orbit" its target or to move in a direction perpendicular to its line of sight to the target. This behavior can be used to model a moth orbiting a flame or a soldier walking sideways while looking at or shooting a target. The *Go between* behavior allows a character to get in between a first target and a second target, similar to a bodyguard coming between its charge and everybody else. The *Flock with* behavior is used to combine the effects of the other three group behaviors, namely the *align with, join with,* and *separate from* behaviors.

These behaviors are additive in that an autonomous character may have multiple active behaviors and in that AI.implant calculates the final motion by combining the component behavioral forces based on each behavior's priority and intensity.

TABLE 1: AI.implant Behaviors

Behavior Type	*Behaviors*
Simple	Avoid barriers
	Avoid obstacles
	Accelerate at
	Maintain speed at
	Wander around
	Orient to
Targeted	Seek to
	Flee from
	Look at
	Strafe
	Go between
	Follow path
	Seek to via network
Group	Align with
	Join with
	Separate from
	Flock with

Action selection and decision trees. The behaviors provided by AI.implant are low-level behaviors that control states or motion of the character with no real thought processes or constraints, except for those provided by the specified parameters for each behavior. Action selection is the mechanism through which AI.implant provides the capability for the user to program more complex behavior for the autonomous character. These decisions can "modify behaviors, drive animation cycles, control the character's state, or update its internal memory" and thus allow the character to sense and react to its environment.

This more complex behavior is achieved by combining multiple features provided by AI.implant™. These include sensors, which allow the autonomous character to gain information about the world; character data, which act like internal memory for a character and which can be modified by sensors or by commands within a decision tree; commands, which are the actual directive to activate or deactivate behaviors or animation cycles, to set character attributes, or to set datum values; decisions, which consist of a conditional expression and a list of commands to invoke; and decision trees, which bring them all together and allow one to nest multiple conditionals together to form more sophisticated logical structures, such as finite-state machines.

Crowd-member character development. For the Mogadishu scenario, we developed three types of characters in our Crowd Federate: Somali citizens (the crowd members), Somali militiamen, and U.S. soldiers. The behavior of the Somali citizens and Somali militiamen are modeled using AI.implant's decision-tree capability. They have similar behavior. They have certain states, defined by their corresponding "state" character data, which controls what they do. They both start in the "init" state, which is where all of its character data are initialized, such as the group that it belongs to, the group that it considers friend, and the group that it considers enemy. We have defined three groups, NeutronGrp, which is our neutral group; ElectronGrp, which is the U.S. soldiers group; and Pion, which is the Somali militia group. The Somali citizens are in the NeutronGrp, with the Pion group being its friend and the ElectronGrp its enemy. All Somali militiamen are placed in the Pion group, and all JSAF objects that appear through the HLA interface are placed in the ElectronGrp.

Other than the initial state, we have identified five other basic states that the Somalis could be in—wander, fight, flee, death, and idle. The Somali citizen has the wander, flee, and idle states defined for it; the Somali militiaman has all five. One important behavior observed during the Mogadishu incident was the use of burning tires to signal the population, when both the militiamen and the noncombatant citizens were likely to converge on the burning tires. The militiamen's goal was to prepare themselves to engage the enemy (here, the U.S. soldiers). The citizens were there to disrupt the enemy's mission by providing cover for the militia, setting up roadblocks, and, in general, causing confusion and mayhem. To account for this behavior, we have coded it for both Somali citizen and militiaman in their wander state. During this state, the resultant action is computed from 3 main behaviors—*Wander around, Flock with* members of its own group, and *Seek to,* with the tire as the target. Therefore, if a character is alone, it will wander around but always toward the tire because of the active *Seek to* behavior. If there are other members of its group nearby, that character will also join and align with those characters in a *Flock with* behavior.

About half of the crowd members will wander around, but the other half will be standing idly by, to show that citizens may be walking or milling around during the course of the simulation. The Somali citizen was also coded with the behavior of fleeing away from danger or, in our case, fleeing from any of the U.S. military forces. On the other hand, the Somali militiaman will not flee from, but will engage and try to "fight," the U.S. soldier. In the prototype, this engagement is in the form of shooting at the soldiers with guns. Finally, the Somali militiaman can die, or enter the "death" state, when it is shot at by any JSAF entity.

If we scan the decision trees (see Figures 7 and 8), we can see the complexity of the types of behaviors that can be implemented. At the top, we have the decision-tree node, which is a decision node. Because the decision type is "always," this branch of the tree will always be executed. From there, we see that several statements follow that will be executed every time this branch is traversed. These statements are the *activate avoidBarrierB1, activate avoidObstaclesB, set targetD = nearestEnemy SoldierD* statements. From these few commands, we see examples of the different AI.implant hooks provided for complex behavior. First, we have the *activate* "behavior," which is an example of a command that allows the user to activate or deactivate specified behaviors based on the scenario. The a*voidBarrierB1* and *avoidObstaclesB* are particular instances of the predefined low-level behaviors, avoid barriers and avoid obstacles, provided by AI.implant. So when this decision tree is encountered for a particular character, that character will always have its avoid barrier and avoid obstacle behaviors activated. The next statement, *set targetD,* is an example of the use of sensors and character data. Here, the datum being modified is the *targetD*

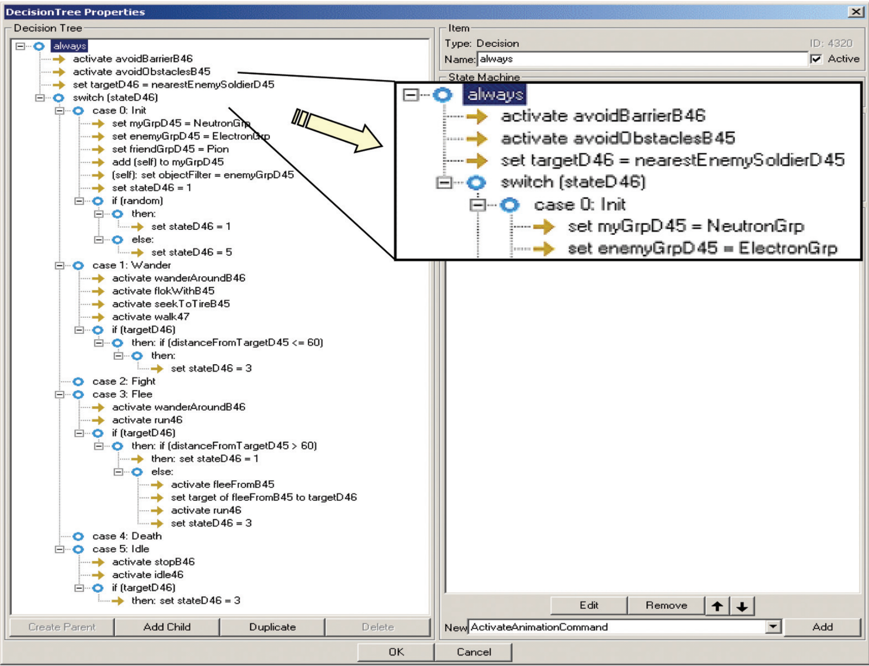

FIGURE 7: Somali Citizen Decision Tree

datum, which stores in the character's memory the identity of the target. The datum *nearestEnemySoldierD* is the result of a visual sensor that "sees" the nearest enemy solider and sets that value in that datum.

With that understanding, we can now walk through a decision tree, which is the decision tree for a Somali militiaman. We see that on executing this decision tree, the first actions taken are to activate the avoid barrier and avoid obstacles behaviors, and to set the *targetD* datum to the closest enemy. Next is a *switch* decision node, which determines which branch of this decision tree to continue down based on the value of the *stateD* datum stored for that character. If the value stored on that datum is equal to 0, then we interpret that as being in the "Init" state, which is the state used to initialize various data. In that state, the following occurs to initialize the character:

1. Set *myGrpD* datum to Pion, which is the Somali militia group.
2. Set *enemyGrpD* datum to ElectronGrp, which is the U.S. soldiers group.
3. Set *friendGrpD* datum to NeutronGrp, which is our citizen group.
4. Add myself to the militia group pointed to by *myGrpD*.
5. Initialize my objectFilter to filter for entities belonging to the enemy group pointed to by *enemyGrpD*.
6. Set *fireD* to 0, which means that I am not firing.
7. Set my *stateD* to 1, which takes me into the Wander state.

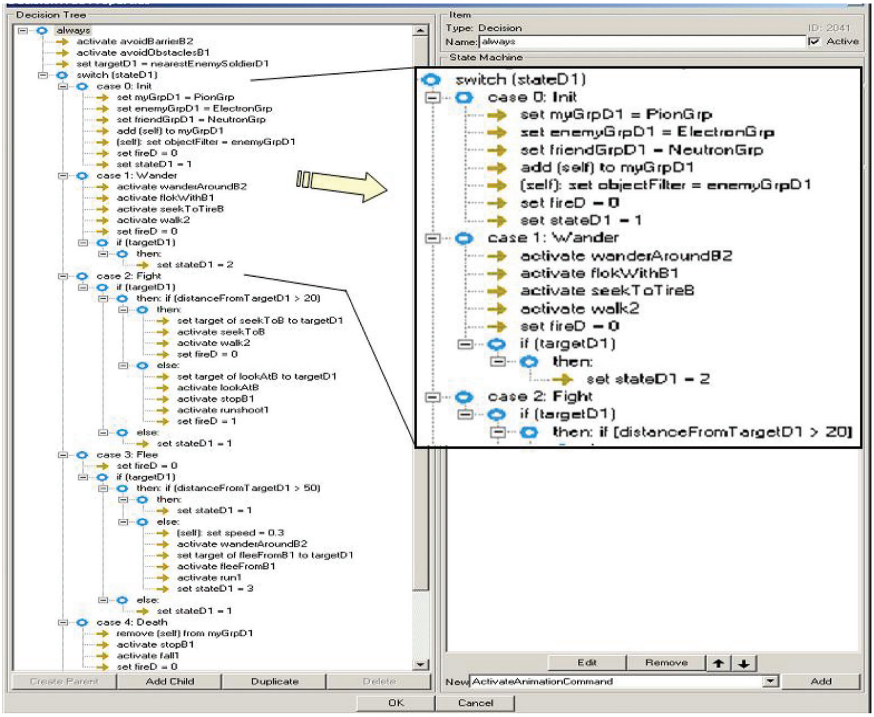

FIGURE 8: Somali Militiaman Decision Tree

As can be seen in the decision tree, after the initialization state has been executed, the character can go into any of the other states that have been identified, such as the wander, fight, flee, and so forth. In each of these states, different actions are scripted to occur, with a more detailed look into those actions further detailed in the next section. The important thing to understand is that the decision tree can be made as simple or as complex as needed for any particular scenario.

The U.S. soldier is modeled as a *player-controlled character* in AI.implant, which is one when there are no behaviors associated with the character. Figure 9 shows the window used to edit decision-tree behavior in AI.implant. The area of concern in the figure is the left-hand panel of the window. Near the top of the panel is a switch statement that determines the case or state of a character. The rest of the decision tree acts like a finite state machine, executing behaviors or actions associated with the particular state of the character. The finite states of the U.S. Soldier character are primarily associated with setting flags or activating animation clips. There are no autonomous behaviors associated with the character because all of its movement, orientation, speed, and so forth are controlled by an external source.

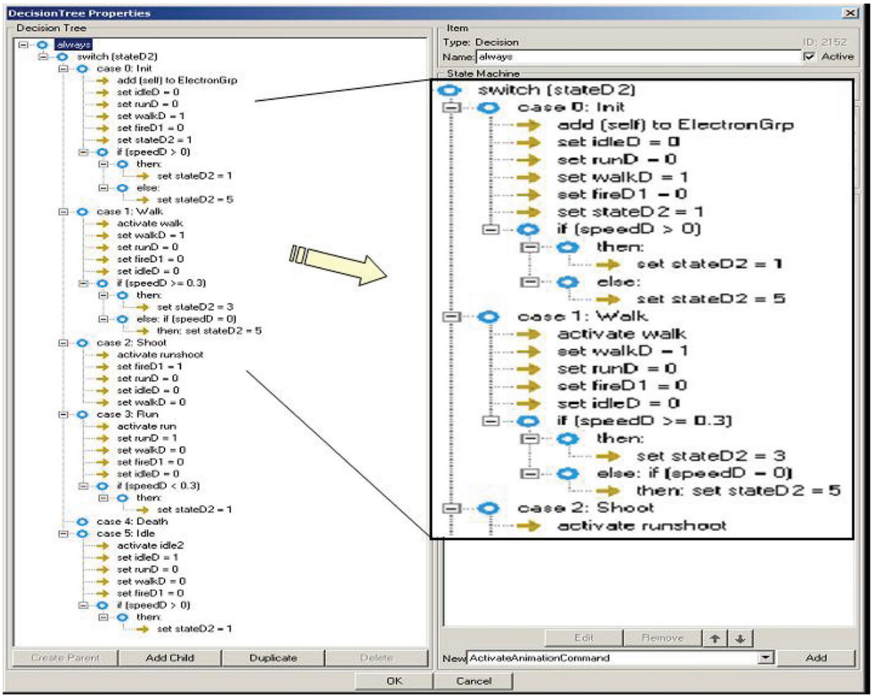

FIGURE 9: U.S. Soldier Decision Tree

The external source for our prototype is JSAF, which was selected to do what it does best, simulate military forces. JSAF is in charge of modeling the behavior of the U.S. soldiers, and sends the updates of the soldiers' entity state such as location, orientation, and speed through its HLA interface.

Normally, we would have used AI.implant's non-autonomous player-controlled character to model the JSAF entities. However, our desire to also incorporate a 3D viewer made this choice impractical. Although we didn't want AI.implant to control any of the movement and behavior associated with the JSAF entities, we did need it to control the animation of the characters for the Arena viewer.

In order to accommodate this requirement, we had to modify the implementation of the player-controlled character. Instead of using the non-autonomous character, we had to use an autonomous character that had no behavior associated with it. We developed a decision tree that was devoid of all behaviors, and had only commands that activated the appropriate animation clips for our 3D viewer.

3D visualization

A 3D viewer was needed to provide a three-dimensional visualization of the executing scenario. This viewer was required to visualize both the crowd entities and the military units in the context of the scenario's terrain database. The 3D HLA viewer that was developed used the BGT Arena engine that is based on the Criterion Software RenderWare game rendering middleware libraries. We incorporated into the supplied Arena viewer an HLA interface to make it interoperable with the Crowd Federate and JSAF. In our prototype, we were able to demonstrate two-way communication between our Crowd Federate and JSAF. We were able to create crowd members in our federate, publish it to the RTI, and have those characters be recognized and shown in the JSAF Panel View Display (PVD). Likewise JSAF entities were visualized within the Crowd Federate.

When we integrated the 3D viewer with HLA, we noticed some peculiar behavior exhibited by JSAF entities. We found that when we gave some JSAF entities the order to move in a direction opposite to the way they were facing, they would not change their orientation before moving in that direction. In fact, the soldiers that were used for our scenario were marching backward! This is not so obvious when you view those same entities in the JSAF PVD, because it is a 2D display, but it became obvious in the 3D viewer. The ability to observe these and possibly other nonintuitive behaviors made incorporating a 3D viewer an important component of our federation.

User experimentation with the Crowd Federate

We had success integrating game technology with military technology to create a system called the Crowd Federate that could interoperate with other military simulations to provide a realistic environment involving interactive civilian crowds. However, the usefulness of such a tool was still in question until we were able to deploy the Crowd Federate to the Army Maneuver Support Battle Lab at Ft. Leonard Wood, Missouri, for employment in a user training/experiment setting. The Crowd Federate was employed in the Maneuver Enhancement Brigade (MEB) Study #1 Exercise. This exercise involved several SAF simulations, several platform simulators, and military police trainees and/or role players. The objectives of the Crowd Federate involvement in the exercise were to

1. demonstrate Crowd Federate interoperability with OneSAF TestBed SAF (OTBSAF),
2. provide crowd entities for employment in the MEB study, and
3. demonstrate capabilities unavailable using on-site tools alone.

Experiment runs conducted

The study was conducted as a series of experiments (or runs). The Crowd Federate was exercised in various scenarios and demonstrated capabilities that were not available organically in the military simulations in use. Figure 10 shows a screen shot of the Crowd Federate 3D Viewer illustrating the typical crowds employed during the experiment runs. The runs conducted during the exercise that involved the use of the Crowd Federate are outlined below along with the crowd size generated.

FIGURE 10: MSBL Deployment Renderer Screen Shot

Run 1: Crowd size: 35. In the first run, a crowd of 35 civilians was generated. Interoperability with OTBSAF was demonstrated that achieved the first objective.

Run 2: Crowd size: 100. In this run, a crowd of 100 civilians was employed. To the delight of the users, the crowd stopped the progress of an OTBSAF vehicle, thereby achieving a capability previously unavailable. Figure 11 shows this capability, which was screen captured from an OTBSAF PVD. The rectangles are trucks in a convoy traveling down the road and the dark specks are the crowd members.

Run 3: Crowd size: 100. In this run, a crowd of 100 civilians was generated. There was generally about a 45-minute break taken between runs to allow for exercise reset. Also, between runs, various modifications to the crowd scenario were requested. For this run, crowd-scenario modification and restart time was reduced to less than 30 minutes.

Run 5: Crowd size: 100. In this run, a crowd of 100 civilians was generated. Unique compared to the previous runs, the Crowd Federate interoperability with one of several vehicle trainers (platform simulators) was demonstrated.

Run 7: Crowd size: 200. In this run, a crowd of 200 civilians was generated. Crowd Federate interoperability with OTBSAF was demonstrated by interaction with military police entities. The interaction resulted in military police firing into the crowd, resulting in an increase in crowd aggression level. The OTBSAF entities sensed the crowd entities initially as neutral, then as hostile. This was also a welcome new capability for the users, as they were able to accomplish an exercise master scenario event list (MSEL) item.

Results

All of the objectives outlined for the exercise were achieved. Crowd Federate interoperability with OTBSAF was demonstrated, and the crowd entities were able

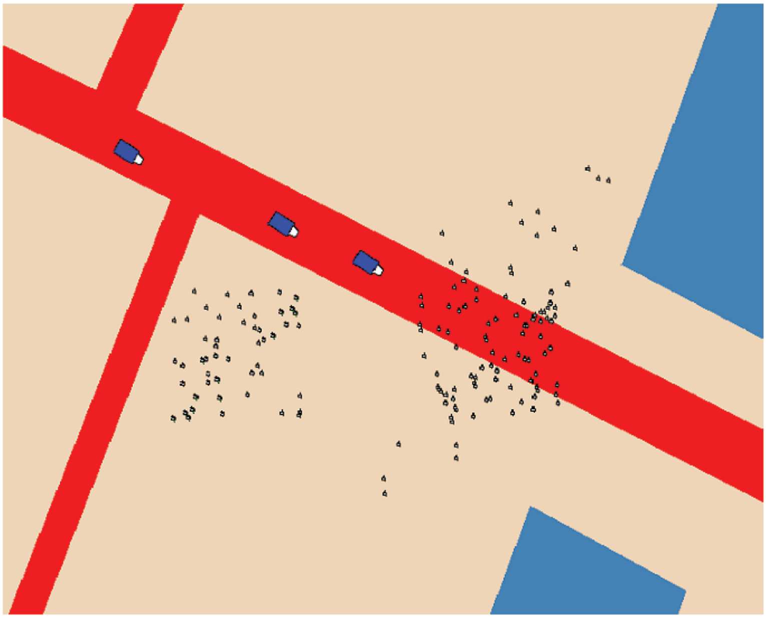

FIGURE 11: Crowd Federate Crowd Blocking the Passage of an OTBSAF Convoy

to be employed in the MEB study. In addition, capabilities that were not previously available to the users were demonstrated. These included the blocking of an OTB-SAF convoy and the transformation from friendly civilians to enemy targets as perceived within the SAF's. Another achievement by our standards was the on-site "markup" of the terrain and the rapid modifications to the scenario in support of the user's requirements.

Summary

VMASC is developing a crowd-modeling capability for military simulation. The first phase of the project consisted of a requirements analysis to identify military simulation crowd-modeling requirements, a literature survey to examine psychological research relevant to crowd modeling, and a design study to explore design issues in the implementation of a crowd simulation. The second phase, currently ongoing, is centered on the development of a crowd simulation implemented as a distributed simulation federate. The Crowd Federate is interoperable with existing military simulations, contains a reconfigurable architecture, and will have a credible psychological basis for the crowd behavior that it generates.

This research drove the convergence of military simulation technology with gaming simulation technology. By doing this, we are able to produce not only a highly realistic simulation but also one with easily reconfigurable Game AI behaviors, which is needed for psychologically based models that are evolving based on changing knowledge, changing context, and the influences of culture. The federate also has the capability of displaying the simulation in a highly detailed manner by taking advantage of gaming-technology rendering engines, which are optimized for visual quality and speed performance. The usefulness of such an application was demonstrated in the support of a real-user exercise at the Army Maneuver Support Battle Lab at Ft. Leonard Wood, Missouri. The combination of flexibility and visual realism makes the use of game technology in military simulation a practical marriage that should be exploited well into the future.

Notes

1. These application areas were used: training, analysis, experimentation, and acquisition.
2. These levels of warfare were used: tactical, operational, and strategic.

References

Air Force Research Laboratory/Human Effectiveness Directorate/Directed Energy Bioeffects Division. (2003). *Crowd control variables working session notes*. Presentation at AFRL/HEDR 2nd Crowd Control Modeling and Simulation Workshop, San Antonio, TX.

Aubel, A., Boulic, R., & Thalmann, D. (2000). Real-time display of virtual humans: Levels of details and impostors. *IEEE Transaction on Circuits and Systems for Video Technology, 10*, 201-217.

Banarjee, S., Grosan, C., & Abraham, A. (2005, September 25-29). Emotional ant based modeling of crowd dynamics. *Proceedings of the Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC '05)*, pp. 279-286.

Bandini, S., Vizzari, G., & Manzoni, S. (2004, June 14-17). Situated Cellular Agents for Crowd Simulation and Visualization. *Transactions of the 2nd Biennial Meeting of the International Environmental Modelling and Software Society (iEMSs),* pp. 289-294.

Bio|Graphic Technologies. (2004, August). AI.implant™ home page. Retrieved from http://www.biographictech.com/

Bliss, J. P., Fallon, C. K., Headen, E. N., & Gaskins, R. (2004, September 20-24). Crowd reactions to sublethal weapons: Universal triggers for crowd violence. *Proceedings of the 48th Annual Meeting of the Human Factors and Ergonomics Society*, *New Orleans, LA.* Santa Monica, CA: Human Factors and Ergonomics Society.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: From natural to artificial systems.* New York: Oxford University Press.

Bouvier, E., & Guilloteau, P. (1996, February). Crowd simulation in immerse space management. *Proceedings of Eurographics Workshop on Virtual Environments and Scientific Visualization '96,* pp. 104-110.

Bowden, M. (1999). *Black Hawk down: A story of modern war*. New York: Atlantic Monthly Press.

Brogan, D., & Hodgins, J. (1997). Group behaviors for systems with significant dynamics. *Autonomous Robots, 4*, 137-153.

Chandrasekaran, B., & Josephson, J. R. (1999, May 11-13). Cognitive modeling for simulation goals: A research strategy for computer-generated forces. *Proceedings of the Eighth Conference on Computer Generated Forces and Behavioral Representation, Orlando FL*, pp. 117-126.

Filippidis, L., Galea, E., Gwynne, S., & Lawrence, P. (2006). Representing the influence of signage on evacuation behavior within an evacuation model. *Journal of Fire Protection Engineering, 16*(1), 37-73.

Kenny, J. M., McPhail, C., Farrer, D. N., Odenthal, D., Heal, S., Taylor, J., et al. (2001, January 1). *Crowd behavior, crowd control, and the use of non-lethal weapons* (Technical report). University Park: Pennsylvania State Applied Research Laboratory.

Kuhl, F., Weatherly, R., & Dahmann, J. S. (1999). *Creating computer simulation systems.* Englewood Cliffs, NJ: Prentice Hall.

Lim, L., & Thalmann, D. (2002, August). Construction of animation models out of captured data. *Proceedings of the IEEE International Conference on Multimedia and Expo '02*, *1*, pp. 829-832.

McKenzie, F. D., Garcia, H., Nguyen, Q. H., Seevinck, J., Xu, Q., & Petty, M. D. Mogadishu terrain generation and correlation for crowd modeling. (2004, April 18-23). *Proceedings of the Spring 2004 Simulation Interoperability Workshop, Arlington, VA*, pp. 944-950.

McKenzie, F. D., Xu, Q., Nguyen, Q. H., & Petty, M. D. (2004, September 19-24). Crowd Federate architecture and API design. *Proceedings of the Fall 2004 Simulation Interoperability Workshop,* pp. 574-587.

McPhail, C., Powers, W. T., & Tucker, C. W. (1992). Simulating individual and collective actions in temporary gatherings. *Social Science Computer Review, 10*(1), 1-28.

Moore, T. (1990). Keep it cool! *Police, 22*(11), 32-33.

Musse, S. R., Garat, F., & Thalmann, D. (1999, September 7-8). Guiding and interacting with virtual crowds in real-time. *Proceedings of the Eurographics Workshop on Computer Animation and Simulation '99,* pp. 23-34.

Musse, S. R., & Thalmann, D. (2001). Hierarchical model for real time simulation of virtual human crowds. *IEEE Transactions on Visualization and Computer Graphics, 7*(2), 152-164.

O'Sullivan, C., Cassell, J., Vilhjalmsson, H., Dobbyn, S., Peters, C., Leeson, W., et al. (2002). Crowd and group simulation with levels of detail for geometry, motion and behavior. *Proceedings of the Third Irish Workshop on Computer Graphics,* pp. 15-20.

Petty, M. D., McKenzie, F. D., & Gaskins, R. C. (2003, October 29-31). Requirements, psychological models, and design issues in crowd modeling for military simulation. *Proceedings of the Huntsville Simulation Conference 2003,* pp. 237-245.

Petty, M. D., McKenzie, F. D., Gaskins, R. C., & Weisel, E. W. (2004, April 18-23). Developing a Crowd Federate for military simulation. *Proceedings of the Spring 2004 Simulation Interoperability Workshop,* pp. 483-493.

Reynolds, C. W. (1987, July 27-31). Flocks, herds, and schools: A distributed behavioral model. *Proceedings of SIGGRAPH '87,* pp. 25-34.

Rysewyk, L. A. (1994). *Experiences of executive officer from Bravo Company, 3rd Battalion, 75th Ranger Regiment and Task Force Ranger during the Battle of the Black Sea on 3-4 October, 1993 in Mogadishu, Somalia.* Fort Benning, GA: U.S. Army Infantry School, Combined Arms and Tactics Division.

Seo, H., Yahia-Cherif, L., Goto, T., & Magnenat-Thalmann, N. (2002, June 19-21). Genesis: Generation of e-population based on statistical information. *Proceedings of Computer Animation 2002*, pp. 81-85.

Tecchia, F., Loscos, C., & Chrysanthou, Y. (2002). Image-based crowd rendering. *IEEE Computer Graphics and Applications, 22*(2), 36-43.

Ulicny, B., & Thalmann, D. (2001, September 2-3). Crowd simulation for interactive virtual environments and VR training systems. *Proceedings of Eurographics Workshop on Animation and Simulation*, pp. 163-170.

Ulicny, B., & Thalmann, D. (2002). Towards interactive real-time crowd behavior simulation. *Computer Graphics, 21*(4), 767-775.

Varwell, D. W. P. (1978). *Police and public.* Plymouth, UK: MacDonald and Evans.

Frederic (Rick) D. McKenzie is an associate professor of electrical and computer engineering at Old Dominion University, where he currently serves as principal investigator (PI) and co-PI on projects

involving medical modeling and simulation, behavior representation in simulations, and simulation architectures. Prior to joining ODU, he held a senior scientist position at Science Applications International Corporation (SAIC), serving as PI for several distributed simulation projects. At SAIC, he was a team lead on a large distributed simulation system. He has more than 10 years of research and development experience in the software and artificial intelligence fields, including object-oriented design and knowledge-based systems. Both his MS and PhD work have been in artificial intelligence, focusing on knowledge representation and model-based diagnostic reasoning.

Mikel D. Petty is director of the University of Alabama in Huntsville's Center for Modeling, Simulation, and Analysis. He received a PhD in computer science from the University of Central Florida in 1997. Dr. Petty has worked in modeling and simulation research and development since 1990 in areas that include simulation interoperability and composability, human behavior modeling, multi-resolution simulation, and applications of theory to simulation. He served on a National Research Council committee on modeling and simulation, is a Certified Modeling and Simulation Professional, and is an editor of the journals SIMULATION *and* Journal of Defense Modeling and Simulation.

Paul A. Kruszewski is the president and founder of GRIP Entertainment (GRIP), an independent Montreal-based game development company. Prior to GRIP, he was the founder of BioGraphic Technologies (BGT), the creator of the artificial intelligence video game technology AI.implant which was sold into some of the world's largest video game publishers, including Electronic Arts and Vivendi Universal Games. In 2005, BioGraphic was acquired by Engenuity Technologies, a Montreal-based simulation company where Dr. Kruszewski became chief technology officer. He received a PhD in computer science from McGill University in 1996, and he has worked in real-time procedural modeling and animation since 1992 in areas that include natural branching pattern simulation (trees, lightning, lungs), procedural human generation, procedural clothing generation, and AI-driven animation.

Ryland C. Gaskins III is a senior research scientist at the Virginia Modeling, Analysis and Simulation Center at Old Dominion University. He received a PhD in human factors psychology from George Mason University in 1995. Since joining the ODU faculty, he has taught a wide range of graduate and undergraduate psychology courses. His research interests are in the area of training and simulation, virtual environments, macro-ergonomics, persuasive computing, human abilities, and task characteristics measurement for selection and placement.

Quynh-Anh (Mimi) H. Nguyen is a graduate research assistant and PhD student in the Engineering Modeling & Simulation Program at the Virginia Modeling, Analysis and Simulation Center at Old Dominion University. She received her ME degree in engineering modeling and simulation from Old Dominion University in 2003 and her BS in electrical engineering from George Mason University in 1997. Prior to coming to ODU, she held a staff engineer/scientist position at Lockheed Martin Naval Electronics & Surveillance System–Undersea Systems, where she worked on several modeling and simulation projects, including one focused on developing a collaborative engineering environment and another developing complex system-performance models using discrete-event simulations. While at VMASC, her research has focused on using military simulation and gaming technology to develop a Crowd Federate and crowd cognitive model.

Jennifer Seevinck is a research scientist at Virginia Modeling, Analysis and Simulation Center (VMASC) East's virtual reality facility. Here she has worked on medical and engineering visualizations as well as the virtual environment training project. Her work at ODU includes object modeling/texturing/animation and project managing. Prior to joining VMASC in 2001, she established and lectured in computer animation and multimedia at Australia's Deakin University. She has held similar faculty positions at the Australian National University and worked as a freelance designer on interactive museum exhibits, Web, *architecture, theater, and film. She has an undergraduate degree in architectural design and a master's in electronic arts with a thesis in virtual-reality interface design from the Australian National University.*

Eric W. Weisel is president of WernerAnderson, Inc., a private, veteran-owned small business that performs research and development in the areas of defense-related operations research and simulation. He received a PhD from Old Dominion University in 2003, an MS in operations research from the Florida Institute of Technology in 1995, and a BS in mathematics from the United States Naval Academy in 1988. His research interests include simulation formalism and composability. He previously served as a U.S. Navy submarine officer with experience in nuclear engineering, operations management, navigation, and joint operations.

ADDRESSES: FDM, RCG, and JS: Virginia Modeling, Analysis and Simulation Center, Old Dominion University, Norfolk, VA 23529, USA; telephone: +1 (757) 683-5590; fax: +1 (757) 683-5590; e-mail: fmckenzi@ece.odu.edu, rgaskins@odu.edu, jseevinc@odu.edu, PAK: GRIP Entertainment, #150-4333 Ste. Catherine Ouest, Montreal, Quebec, Canada H3C 1P9; e-mail: paul.kruszewski@gripentertainment.com. Q-AHN, EWW, WernerAnderson, Inc., 6655 Main Street, P.O. Box 118, Gloucester VA 23061, USA; e-mail: qnguyen@werneranderson.com, eweisel@werneranderson.com. MDP: University of Alabama in Huntsville, Von Braun Research Hall D-14, 301 Sparkman Drive, Huntsville AL 35899, USA; e-mail: pettym@uah.edu.