The Application of Commercial Gaming Technology to Adaptive Adversarial Behaviors

Benjamin D. Hamilton
Technical Support Working Group (SETA CTR)
Arlington, VA
hamiltonb@tswg.gov

Rodney Long
RDECOM-STTC
Orlando, FL
rodney.long@us.army.mil

Nicole Coeyman
RDECOM-STTC
Orlando, FL
nicole.coeyman@us.army.mil

Vivian Gottesman
L3 Communications
Orlando, FL
vgottesman@link.com

Jon Williams
L3 Communications
Orlando, FL
jmwilliams@link.com

ABSTRACT

As computer gaming technology continues to improve it has come to rival or surpass the simulated imagery, dynamics, and human behavior representation available in current military training simulators. With the goal of applying gaming technologies to training simulations, the Technical Support Working Group (TSWG), through the U.S. Army Research, Development and Engineering Command (RDECOM) Simulation and Training Technology Center (STTC), has sponsored an effort to use a commercial game engine for the simulation of fully automated and adaptive individual adversaries. This paper discusses the use of gaming technology to implement fully automated and adaptive adversarial behaviors. The use of an AI gaming engine allows the adversarial behaviors to adapt by assessing local conditions and dynamically changing tactics, target selection and routing. Learning takes place, and tactics improve, during scenario execution and this learning is retained across scenario runs so that an adversary will improve each time a scenario is run. AI.implant, a commercial artificial intelligence game engine, was interfaced with the behavioral architecture of OneSAF Test Bed (OTBSAF) to provide OTBSAF simulated entities with adaptive adversarial behaviors. Several behaviors were implemented, including a Suicide Bomber and an IED Ambush. A graphical user interface was developed that provides the non-programmer the ability to modify existing behaviors or to create entirely new behaviors. An evaluation was conducted to assess the effectiveness of the AI.implant behavior engine in terms of variability, adaptability and autonomy. Additionally subject matter experts (SMEs) were used to evaluate the ease with which a non-programmer can create or modify adaptive behaviors. The results of these evaluations are provided and discussed. The developed adversarial behavioral system is currently in a usable state, and work to interface the system with existing training systems is discussed.

ABOUT THE AUTHORS

Benjamin D. Hamilton is a Senior Program Analyst and SETA Support Contractor to the Technical Support Working Group. His expertise includes analyzing, designing, evaluating, and managing projects involving interactive instruction, performance improvement, and simulation/game-based training. He received his Masters degree in Instructional Technology from Bloomsburg University and his Doctorate of Education in Instructional Technology and Distance Education from Nova Southeastern University.

Rodney Long is a Science and Technology Manager at the SFC Paul Ray Smith Simulation and Training Technology Center (STTC) in Orlando, Florida. The STTC is part of the U.S. Army Research, Development and Engineering Command (RDECOM). Mr. Long is currently the program lead for research using Massively Multiplayer On-Line Games (MMOG) for training asymmetric warfare in urban environments. Mr. Long has a breadth of simulation and training experience that spans more than 20 years in the Department of Defense. Mr. Long has a Bachelor's Degree in Computer Engineering from the University of South Carolina and Master's degree in Industrial Engineering from the University of Central Florida.

Nicole Coeyman is a Science and Technology Manager for the Asymmetric Warfare - Virtual Training Technologies (AW-VTT) program at the U.S. Army Research Development and Engineering Command (RDECOM) Simulation and Training Technology Center (STTC). Her research interests include virtual environments, Massively Multiplayer On-Line game (MMOG) technologies, robotics, simulation technologies, and digital systems. She received her Bachelor's Degree in Computer Engineering from the University of Central Florida, and is currently pursuing a Master's Degree in Industrial Engineering (Engineering Management) from the University of Central Florida.

Vivian Gottesman is employed by L3 Communications as a Senior Software Engineer. She received her BS in Mathematics with minors in Computer Science and Physics from The Florida State University, Tallahassee, FL. She received her MS in Mathematical Science with an emphasis in matrix theory from The University of Central Florida, Orlando, FL

Jon Williams is employed by L3 Communications as a Principle Systems Engineer. He has 25 years of experience in simulation, including 10 years in rotorcraft engineering development simulation and 15 years of modeling experience with Computer Generated Force systems.

The Application of Commercial Gaming Technology to Adaptive Adversarial Behaviors

Benjamin D. Hamilton
Technical Support Working Group (SETA CTR)
Arlington, VA
hamiltonb@tswg.gov

Rodney Long
RDECOM-STTC
Orlando, FL
rodney.long@us.army.mil

Nicole Coeyman
RDECOM-STTC
Orlando, FL
nicole.coeyman@us.army.mil

Vivian Gottesman
L3 Communications
Orlando, FL
vgottesman@link.com

Jon Williams
L3 Communications
Orlando, FL
jmwilliams@link.com

INTRODUCTION

The application of gaming technologies to training simulations is very popular in today's world where gaming technology continues to advance. This type of technology is used to enhance training simulations by adding realism through better graphics, realistic dynamics, and believable artificial intelligence. Training simulations are lacking the capability to realistically represent adversarial agents who can adapt to their environment and its conditions. The Technical Support Working Group (TSWG) sponsored an effort entitled "Adaptive Simulation Agents for Adversarial Behaviors (ASAAB)," which uses a Commercial, off-the-shelf (COTS) game engine to simulate individuals with adversarial behaviors, who are fully automated and adaptive, in a training simulation. This research was performed as a collaborative effort between the US Army Research, Development and Engineering Command (RDECOM) Simulation and Training Technology Center (STTC), L3 Communications Corporation, and the University of Central Florida (UCF) Institute for Simulation and Training (IST).

The result of the ASAAB research effort is a COTS-based software product that is used to produce and implement adversarial agents in a simulation environment. This project was part of an exploratory effort to advance knowledge in the field of adaptive adversarial agents. The ASAAB tool consists of a user interface to design adversarial entities, AI.implant which is the COTS game engine used to instantiate those entities, and an external simulation used to place and guide those entities. AI.implant is an artificial intelligence route planning or "pathfinding" tool. The external simulation currently used is the Army's OneSAF Testbed Baseline (OTB). Adversarial agents are constructed through the user interface, utilizing the AI.implant game engine, externally to OTB. The resulting agents are then integrated into the simulation and can be viewed through the OTB graphical user interface (GUI). The types of adversarial roles that were implemented using this approach include a Suicide Bomber, Sniper, IED Ambusher, Bomb Maker, and Change of Sides.

This project had many goals, one of which was to create an authoring tool for the non-programmer to be able to develop these adversarial agents and represent individual insurgents in a simulation environment. Additional goals included being able to create agents who are autonomous, variable, and adaptable. The use of an AI game engine allows the adversarial agents to adapt to the environment by assessing local conditions and dynamically changing tactics, target selection, and routes. A qualitative evaluation was performed using SMEs to evaluate the accessibility, acceptability, and usability of the tool. A quantitative evaluation was conducted after numerous simulation runs to evaluate the agents' performance against the autonomy, variability, and adaptability design goals. This paper will address the technical approach used to design this tool, and the results of the evaluations conducted.

DESIGN GOALS

The Adversarial Behavior system design targeted four goals for system performance:

1. Fully Autonomous
2. Variability
3. Adaptability
4. Accessibility

The first of these goals was to create behaviors that were fully autonomous, as opposed to the semi-autonomous behaviors commonly found in current computer generated force (CGF) systems. The semi-automated behaviors require the human operator to specify at least some high level goals for an entity, such as a route or destination to move to, or an engagement location. The adversarial behaviors developed through this effort were to be fully autonomous, requiring no action by the operator beyond adding the adversary to the scenario. The second design goal for the adversarial behavior system was to provide variability in the execution of the behavior. Many CGF systems do provide variability in that they use random numbers as part of internal calculations. Running a saved scenario will yield different results each time the scenario is run. This variability can be easily observed during engagements where the hit/miss, and the resulting damage given a hit, of a shot are both determined probabilistically. On a large scale however, a given scenario will generally unfold in a similar manner from run to run with individual entities planning the same routes to their destinations for each run. The goal for variability in the context of this adversarial behaviors effort was to prevent the trainee from "gaming" the system by learning the details of an adversary's behavior to the point where the adversary becomes predictable and therefore easily countered. As the project title implies adaptability was a focus of this effort. Adaptability refers to the ability of the simulated adversary to improve its performance through multiple runs. Adaptation was implemented both during scenario execution and between scenario runs. Accessibility refers to the ability of a non-programmer to create or modify behaviors. To that end a user interface was developed that provided a graphical representation of the adversarial behavior. This graphical representation can be modified or extended by the user.

TECHNICAL APPROACH

The approach used for this project was to identify a commercial, off the shelf, artificial intelligence (AI) engine to be interfaced with an existing computer generated force system. The AI engine would provide additional behavioral capabilities to the CGF system. AI.implant was chosen as the AI engine. Previous work with AI.implant had shown it to be capable of providing the necessary planning and decision algorithms. Also, the ability to leverage the previous experience with the AI.implant application programmer interface (API) factored strongly in the decision on choice of AI engine.

The CGF system chosen was a version of OTBSAF 1.0 modified for use with the Army Aviation Combined Arms Tactical Trainer (AVCATT) helicopter training simulator. A previous Internal Research & Development effort had accomplished a partial integration of AI.implant with OTBSAF and it was desirable to leverage this previous work. The use of the AVCATT version of OTBSAF would also allow the adversarial behavior system to be made immediately available to both the AVCATT and the Flight School XXI training simulations.

System Architecture

The AI.implant libraries and support tools are targeted at the Windows operating system and as such would have been difficult to include directly into the OTBSAF executable. The most flexible architecture was thought to be running OTBSAF and AI.implant on separate CPUs, communicating with each other by means of a network connection. Using this approach the adversarial behaviors computed by the AI engine could be interfaced to any CGF system regardless of the CGF host architecture. The CGF continued to simulate all aspects of the adversarial entity with the exception of the behavior itself. OTBSAF then continues to model physical movement, sensors, weapons and damage, so aside from the adversarial behavior itself, any verification and validation previously resident in the CGF system has been left undisturbed. To interface the CGF system to the AI engine, a new behavioral model was created in OTBSAF to act as a pass through for information to and from the AI engine. A service library was added to the SAF that the pass through behavior uses to query information requested by the AI engine. This service library also implements the protocol used to communicate between OTBSAF and the AI engine.

Supported Roles

The adversarial agent system was designed to be expressive and to support the implementation of a wide range of behavior. In order to bound the scope of the problem, roles were limited to individuals on foot and acting alone. The following five roles were chosen to test the system's expressiveness, and to provide a

system that was immediately useful upon completion of the effort:

1. Suicide Bomber
2. Sniper
3. IED Ambush
4. Bomb Maker
5. Change of Sides

Subject matter experts were interviewed by the University of Central Florida, Institute for Simulation and Training to determine the specific behavior that would be implemented for each of these.

The Suicide Bomber role places an individual in the scenario equipped with a vest bomb. In the fully autonomous mode the Suicide Bomber will wander the local area around his placement in the scenario and search for aggregations of individuals. The search time randomly varies with each run of an exercise. The Suicide Bomber will select among the larger groups of individuals, walk to this area, and detonate the vest bomb. The Sniper role places an individual in an exercise that will walk throughout an urban setting looking for targets of opportunity. When potential targets are discovered, the sniper will seek a position of cover with line of site to the intended target and a suitable route to egress the area. The sniper will engage the target, egress the area, and travel to another area to seek another target. The IED ambush role places an individual in the exercise that will move about an urban environment seeking an area with heavy traffic. The IED ambusher will place an IED, and then seek a concealed area from which to observe the traffic in the area of the placed IED. When several vehicles are within range of the IED, the ambusher will detonate the IED and egress the area. The Bomb Maker will travel through an urban area stopping at various places to collect parts for the bomb that have been placed about. The Change of Sides role adds to the exercise a neutral entity that will become adversarial after witnessing several other neutral entities injured or killed. Once the switch has been made to adversarial, the individual will engage friendly forces.

Behavior Design Tool

A behavior design tool was developed to allow an individual with no knowledge of programming to create or modify adversarial behaviors.

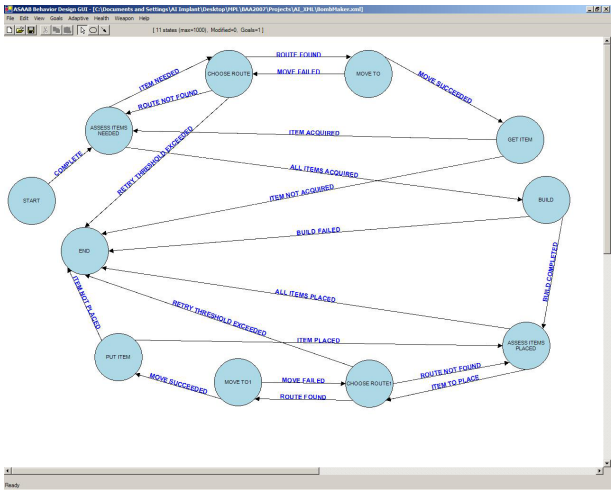

Figure 1 - Adversarial Behavior Design Tool

The design tool is graphical in nature and allows the designer to draw out the behavioral states and state transitions on the computer screen, moving and reconnecting the elements of the state machine until the desired result is achieved. When the graphical representation of the behavior is complete, the tool will provide as output an Extensible Markup Language (XML) file that is consumed by the AI engine to accomplish the actual simulation of the behavior, and a data file that is used by OTBSAF to give the simulation operator access to the adversarial behavior through the SAF user interface.

Variability and Adaptability

It was important to introduce variability into the adversarial behaviors in order to prevent an adversary from becoming predictable, and thus easily countered, as a trainee gained more exposure to scenarios containing these entities. Variability was introduced in three ways. First, the routes chosen by an adversary are selected among many candidate routes provided by the AI engine. The potential routes are analyzed for suitability based on route length and concealment from enemy forces. The adversary will then choose a route randomly from the most suitable 50 percent of available routes. In this way the adversary will avoid choosing poor routes but will not always choose the very best route. In this way the adversary can efficiently make progress toward his goal but will not take a predictable route even though the initial conditions of a scenario have remained the same.

Second, the choice of targets and engagement locations are analyzed and prioritized. For example, a Suicide Bomber will give a crowd of 50 people a higher priority than a group of 10 people since the damage caused by attacking the larger group is likely to be greater, and causing the maximum possible damage is one of the goals of the Suicide Bomber. Again, the adversary will not select the target of highest value, but will select randomly from the top 50 percent, thereby providing a degree of variability.

The third method for introducing variability into the adversarial behavior is to allow the adversary to select its own role from all of the available roles. In this case the operator does not select a role for the adversary, but simply places it in the scenario in the fully autonomous mode. The adversary will analyze the scenario and choose an appropriate role. For example, a scenario that contains a large number of people in groups will give the adversary a preference for taking on the role of Suicide Bomber. A scenario with many vehicles traveling about will give the adversary a preference for the IED Ambush role.
Adaptation was implemented by introducing a set of personality attributes for the adversary such as intelligence, experience, and the current state of resolve, fear and anger. These attributes were tied to specific behavioral actions that directly affect the path taken by the adversary toward the defined goal. The initial settings for each of these attributes are defined by the behavior designer within the Behavior Design Tool. Optionally the designer can give the SAF operator access to these settings at run time through the OTBSAF user interface. The adversarial behavior system self evaluates performance and will vary these attributes to improve performance. The values of these attributes are saved at the end of a scenario and can optionally be used as the initial conditions for them in the next training simulation iteration. In this way the adversarial entity can continue to adapt across scenario runs.

Fully Autonomous

The requirement for fully autonomous operation of the adversarial behavior has been satisfied by allowing the operator to simply place an adversary in a scenario with no other action on the part of the operator. In the fully autonomous mode, the default mode of operation, the adversary will search for and select its own role, targets, and engagement locations. The operator may, at his option, exercise direct control over the adversarial agent, selecting specific roles, routes, choice of weapons, targets and engagement locations.

SAF Modifications

A new behavioral model was implemented in OTBSAF. The user interface for this behavior is changed dynamically based on the selected adversarial behavior. Any parameters that the behavior designer chooses to make accessible to the user are communicated to OTBSAF through the data files output from the Behavior Design Tool. The OTBSAF editor for the adversarial behaviors uses the same user interface elements that are used throughout the rest of OTBSAF, so the interface will be familiar to any OTBSAF operator.

A service library was added to OTBSAF. Its purpose is to establish a non-DIS/HLA communication path with an external application and accept registrations for information needed by the external application, in this case the AI engine. These registrations include a description of the information being requested from OTBSAF and the desired rate at which the information is to be transmitted. Information computed by OTBSAF that has been made available as part of this service are:

1. A list of all entities with a specified range of the requester
2. The force ID of any entity
3. The appearance of any entity
4. Any fire events or detonations within a specified range
5. Adversary position, velocity and orientation as determined by the OTBSAF Lifeform hull model
6. Any entities known to the requesting entity through the use of the OTBSAF modeled sensors installed on the adversary

The AI engine transmits to OTBSAF the desired route of travel and requests that OTBSAF perform any engagements, such as detonating a bomb or firing a sniper rifle.

Terrain Databases

AI.implant uses its own terrain database format called a Navigation Mesh as shown in figure 2.

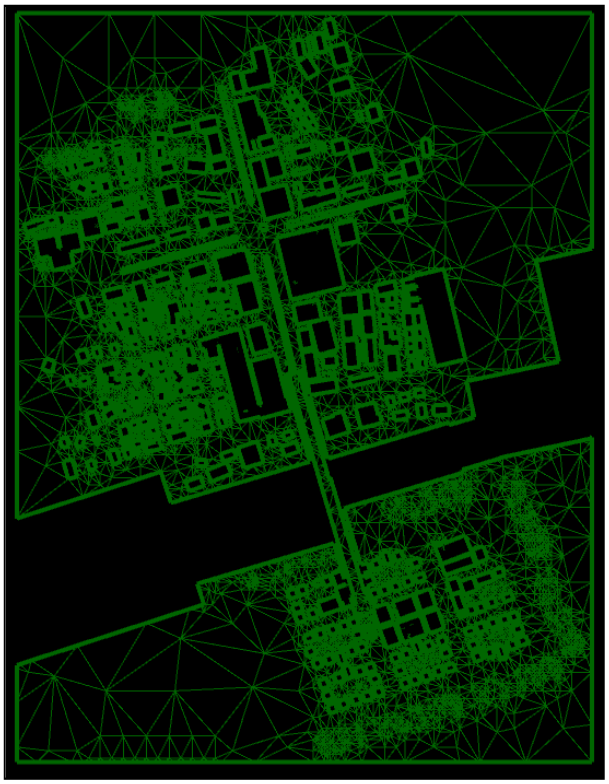

Figure 2 - AI.implant Navigation Mesh

Previous work performed by RDECOM-STTC produced a set of terrain databases that are correlated between AI.implant, OTBSAF, and the On-Line Interactive Virtual Environment (OLIVE) simulation discussed below. These databases represent a portion of Baghdad and were used throughout the development and testing of the adversarial behaviors system. The use of the adversarial behaviors in another geographic area would require the availability of an AI.implant navigation mesh correlated with a Compact Terrain Database (CTDB) for use with OTBSAF.

Interface to OLIVE

The OLIVE is a Massively Multiplayer Online (MMO) training simulation being utilized in developmental projects at RDECOM-STTC. One of the goals of the Adversarial Behaviors effort was to make the adversarial behavior available to the OLIVE simulation. To this end a set of appearance enumerations were defined and visual system animations developed to display each of the following characteristics:

1. No Special Appearance
2. Swaying
3. Nervous
4. Cautious
5. Angry
6. Talking On A Cell Phone
7. Dirty or Dusty
8. Kicking Dirt
9. Digging
10. Photographing or Videotaping
11. Carrying a Shovel
12. Wearing An Armband
13. Carrying a Rifle
14. Wearing a Vest Bomb
15. Carrying a Concealed Bomb
16. Carrying Bomb Parts

Swaying, for example, is characteristic of a person wearing a heavy explosive vest. The trainee would be expected to consider this suspicious and worthy of further investigation. Similarly, a person who is dusty or dirty may have recently been involved in the burying of an IED and may indicate an individual who warrants further attention. An individual who is actively photographing or videotaping an area may be there to document or plan an attack.

TEST METHODOLGY

Testing of the Adversarial Behavior system occurred in two phases. The first phase included a set of quantitative evaluations. In this phase a set of scenarios were developed and an automated scenario execution and data collection facility was used to execute each of these scenarios many times. This automated scenario executions were used to test variability, adaptation and fully autonomous operations. In the second phase of evaluation the Behavior Design Tool and the overall usefulness of the adversarial behaviors for training purposes were evaluated using SMEs. Two types of SMEs were used in this qualitative evaluation: military personnel who act, or who have acted as instructors at training simulation facilities, and personnel who are typical of training simulation operators.

The automated scenario execution and data collection facility collected the following data at regular intervals:

1. Time
2. Entity ID and enumeration
3. Position of each entity in the scenario
4. Health of each entity

5. Fire events
6. Detonation events

Figure 3 shows the quantitative evaluations performed along with the type of adversary used and the number of runs accomplished in each case.

Design Goal	**Cases**	**Entity**		
		Suicide Bomber	Sniper	IED Ambusher
Autonomous	25			X
Variable	100	X		
Adaptive	100		X	

Figure 3 – Quantitative Evaluation Test Matrix

Autonomy

To evaluate agent autonomy, a scenario with an IED ambush was created and run with the adversarial agent starting in five different locations as shown in figure 4.

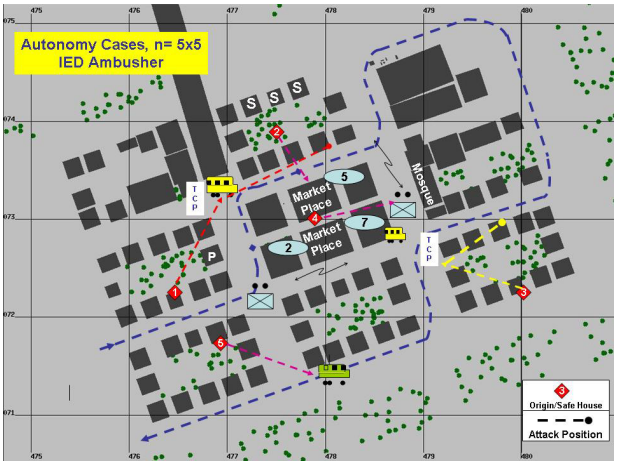

Figure 4 – Autonomy Test Scenario

The scenario was run five times for each of the five starting locations for a total of 25 runs. The criteria for successful autonomous operation of the adversarial agent were for the adversary to automatically:

1. Identify a target
2. Select a route to the engagement area
3. Move to the engagement area
4. Place an IED
5. Conduct the attack
6. Cause casualties

Ninety percent was chosen as the success rate for which the adversary must attack a viable target.

Variability

The variability of adversarial agent behavior was evaluated by running a scenario multiple times with identical initial conditions. The scenario used is shown in figure 5.

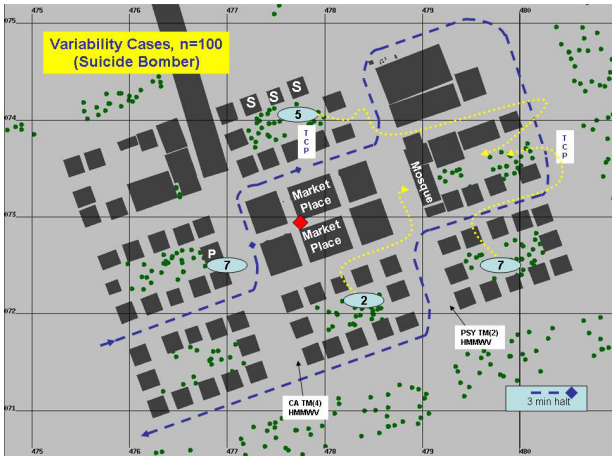

Figure 5 – Variability Test Scenario

The scenario included a suicide bomber and many targets from which to choose. Data was collected on 100 runs with the criteria for success being the selection of a different target for at least 50 of those runs, and a different route selected for at least 50 of the 100 runs.

Adaptation

The Sniper adversarial agent was used to test adaptation. The test scenario defined a fixed starting location, fixed target and fixed engagement location for the sniper. Three versions of the scenario were used. One in which no BLUFOR were present. One in which the BLUFOR moved toward the sniper ingress route from the north, and one in which BLUFOR

moved in from the south. The case with no BLUFOR was run ten times to establish the baseline behavior. The two scenarios that included BLUFOR were each run 45 times. The adversarial agent was to adapt by modifying its routes in order to remain concealed.

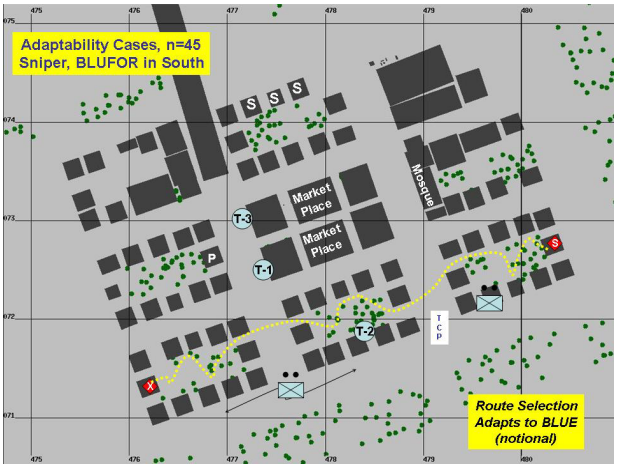

Figure 6 – Adaptation Test Scenario

Accessibility and Usability

Accessibility and usability was evaluated quantitatively using real people to interact with the adversarial behaviors system. Six SMEs were used in this phase of the testing. A training course was developed to familiarize each subject with the capabilities and use of the system and a set of evaluation questions were developed for the subject to answer after interacting with the system.

TEST RESULTS

Autonomy

Of the 25 scenario runs done to test fully autonomous performance, 21 resulted in the detonation of the IED near the intended target, and of those, 12 resulted in the destruction of the target. This gives an 84% success rate, not too far from the 90% goal. The detonations that did not result in a catastrophic kill of the target are attributed to the unclassified damage tables in OTBSAF that are used in making the damage determination.

Variability

Of the 100 variability scenario runs, the Suicide Bomber should not select any single target in more than 50% of the runs, and should also select different routes when the same target was engaged in a least 50% of the runs. The data for the actual runs shows that no single target was selected in more than 38% of the runs. Analysis of the routes taken indicates that the same route was followed in at most 68% of the runs. This is greater than the 50% that had been set as the criteria for success. This may be attributed to the small terrain database used and an insufficient number of choices for route planning.

Adaptation

Of the 90 adaptation scenario runs, the Sniper was expected to adapt the chosen route to evade the moving BLUFOR in 100% of the runs. The sniper and chosen target were set up along an east-west line and a set of runs with no BLUFOR were made to establish the baseline route to the target. Figure 7 shows the north-south movement of the sniper in the absence of BLUFOR.

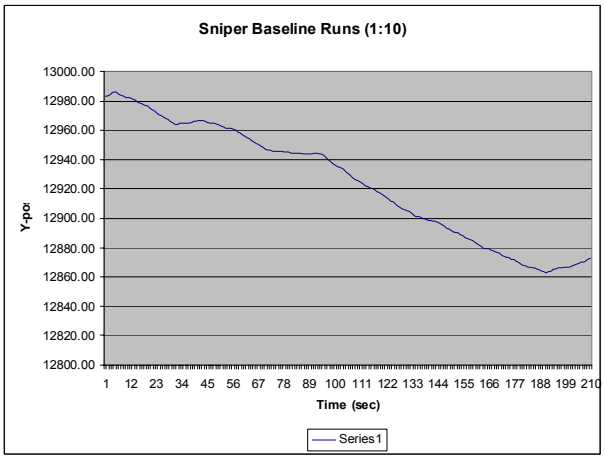

Figure 7 – Baseline Route Data

The test scenario was then run with BLUFOR present and the route data analyzed for north-south deviation from the baseline. Figures 8 and 9 show the deviation of the sniper's route from the baseline route with the BLUFOR moving in from the north and south, respectively. While there is a noticeable deviation in both cases, the deviations are unremarkable. Again, this is likely due to the small terrain database in use for these experiments.

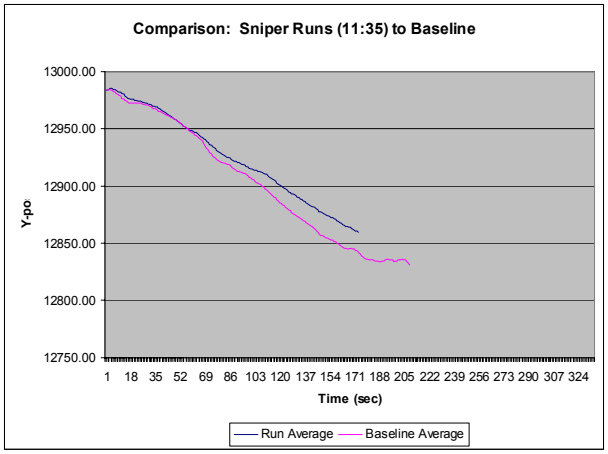

Figure 8 – Route Deviation to the South

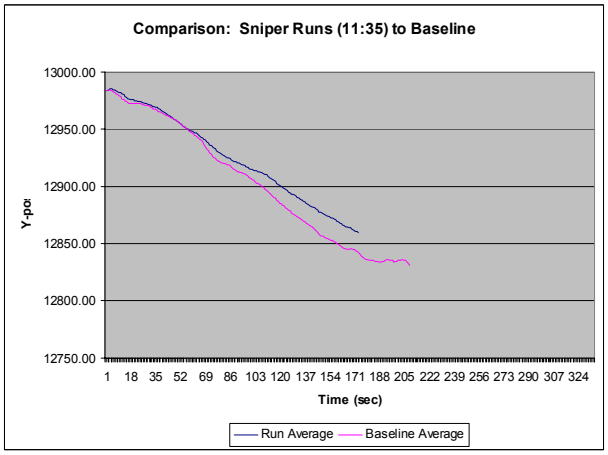

Figure 9 – Route Deviation to the North

CONCLUSIONS

This project successfully integrated a COTS AI engine with an existing CGF system to provide for behavior that is more realistic and less of a burden to the CGF operator. Additionally, the ability to define or modify CGF simulated agent behavior has been made accessible to the non-programmer. From the viewpoint of the SMEs involved with the testing of the Adversarial Agents system, the Behavior Design Tool is the most successful part of this effort. More testing on agent adaptation is warranted and should be done in the context of a more complex environment that will afford the AI engine greater latitude in its planning.

ACKNOWLEDGEMENTS

This research effort was funded by the Technical Support Working Group (TSWG) and was performed by the Government – Industry – Academia (GIA) Simulation Laboratory. The GIA is a consortium under a Cooperative Research and Development Agreement (CRADA) between the RDECOM - (STTC, L-3 Communications Corporation, and the University of Central Florida (UCF) Institute for Simulation and Training (IST). L-3 Communications and the IST performed work for this effort under contracts N61339-07-C-0030 and N61339-07-C-0022, respectively. Forterra Systems, Inc. performed work for this effort under contract N61339-05-C-0004, funded by the TSWG and the RDECOM-STTC.