IMPLEMENTATION RESULTS USING DIFFERENT BEHAVIOR APPROACHES IN THE CGF TEST-BED

Susan A. Gugel, David R. Pratt
Science Applications International Corporation
Orlando, FL

ABSTRACT

A software behavior implements an action of a simulated entity. For example, a behavior can change the stance of an individual combatant (IC) or move an IC to a new position. Currently, behaviors are used to control ICs in many Computer Generated Forces (CGF) simulations such as Closed Combat Tactical Trainer (CCTT) and Modular Semi-Automated Forces (ModSAF). A behavior approach, on the other hand, is a software technique used to implement a behavior. For example, in CCTT and ModSAF the software technique utilizes finite state machines (FSMs). In the past few years, computer hardware technology has provided massive improvements. These improvements combined with the need for more realistic and autonomous behaviors as well as decision-making that handles multitudes of different inputs resulted in the Non-Traditional Human Behavior Models project. The goal was to research several behavior approaches and implement these approaches within a CGF simulation. The project examined in detail traditional FSM, Q-Learning reinforcement, evolutionary, and fuzzy rule-based approaches as each of these approaches provided different mechanisms with different strengths and weaknesses to control ICs in specific use cases. A previous paper was published describing the overall design of the behavior approaches and their relationship to the CGF Test-bed (Gugel, Pratt, & Smith, 2001). This paper, the second in a series, details the scenario (and its variants) selected to evaluate the four behavior approaches. The paper describes the specific scenario design for each approach. The next section describes the results of the experimentation for each approach and scenario variant combination. The final section outlines the overall results across all of the experimentation. This section also outlines overall benefits and weakness of these approaches with respect to implementation. We believe that understanding different behavior approaches and allowing different approaches to exist within the same CGF simulation will allow a diversity of new behaviors to be developed that provide more realism as well as more automation. We believe that these approaches can provide an accurate portrayal of CGFs in training simulations and provide a more versatile simulation for the analysis of new doctrine and tactics.

ABOUT THE AUTHORS

Susan A. Gugel has over seven years of experience in virtual entity simulations utilizing Computer Generated Forces as a Senior Software Engineer at the ASSET Group within SAIC. Over this period, Ms. Gugel has supported direct CGF simulation projects such as the CCTT program, United Kingdom Combined Arms Tactical Trainer (UKCATT) program, and One Semi-Automated Forces (OneSAF) Program. In addition, Ms. Gugel has supported research programs such as Semi-Automated Behavior Generation System (SBGS) and Non-Traditional Human Behavior Models (NHBM). Throughout these programs, Ms. Gugel has concentrated on behavior architectures, implementation, testing and supporting tools. Ms. Gugel received her Master of Science in Computer Science from the University of Central Florida in 1993.

Dr. David R. Pratt is a Chief Scientist/Fellow at SAIC ASSET Group. His active research areas include semi and fully autonomous systems, data management, and language performance evaluation. Before joining SAIC, he has been the JSIMS Technical Director, a tenured Associate Professor of Computer Science at the Naval Postgraduate School, and a Captain in the United States Marine Corps. Dr. Pratt has over 50 publications covering a wide range of computer topics and was the PI on over $5.5M worth of research grants. He currently teaches part time at the School of Computer Science, University of Central Florida. Dr. Pratt received a Ph.D. in Computer Science in 1993 and a Masters of Science degree in Computer Science in 1988 from Naval Postgraduate School.

IMPLEMENTATION RESULTS USING DIFFERENT BEHAVIOR APPROACHES IN THE CGF TEST-BED

Susan A. Gugel, David R. Pratt
Science Applications International Corporation
Orlando, FL

INTRODUCTION

The National Research Council in Modeling Human and Organizational Behavior (1998) has stated that "…future models and simulations used to train military forces, develop force structures, and design and develop weapon systems must be able to create more realistic representations of the command and control process and the impact of command decisions on the battlefield outcomes" (p. 12). As a result of this need, the Non-Traditional Human Behavior Models project examined four different behavior approaches to determine if these approaches would be suitable for modeling and simulation applications. These approaches included hard-codes approaches that enabled the direct coding of military doctrine as well as learning approaches that selected a course of action to follow.

The project included several steps:

1. Identify behavior approaches for implementation
2. Develop a set of common scenarios
3. Select the reused CGF simulation to provide the infrastructure
4. Design and implement the behavior approach engines within the CGF Test-bed
5. Implement the Take Cover under Fire scenario for each behavior approach
6. Conduct experiments using the Take Cover under Fire scenario
7. Continue experimentation based on results

Detail of steps 1-4 can be found in Gugel, Pratt, and Smith (2001) which is the first paper of this series. It details the common design and method for allowing multiple behavior approaches to be utilized with the CGF Test-bed at once. Additional information about the CGF Test-bed can be found in Courtemanche (2000). An overview of the behavior approaches, the Take Cover under Fire scenario, the experimentation, and overall results are presented in the following sections along with suggestions for future experimentation.

OVERVIEW OF BEHAVIOR APPROACHES

The four behavior approaches examined included a Finite State Machine (FSM) approach, a Q-Learning approach, an evolutionary approach, and a fuzzy rule-based approach. The FSM approach which hard-codes the behavior (or at least behavior primitives) in code is the approach used in Modular Semi-Automated Forces (ModSAF) and Combine Arms Tactical Trainer Semi-Automated Forces (CCTT SAF). The Q-Learning and evolutionary approaches are both learning approaches. They use scenario-specific input, such as the actions selectable and the rewards to maximize, across numerous learning iterations to "learn" the correct action to take to in a given situation. The final approach is the fuzzy rule-based approach. The fuzzy approach utilizes fuzzy logic to determine what course of action to take and uses preferences from "losing" courses of action to help set parameters for the selected course of action. The approach uses fuzzy logic, similar to Boolean evaluations except the return values can reside between 0 and 1, to simulate a more human interpretation of the environment. The fuzzy logic is described by terms that have no exact meaning such as *near* and *recent*.

TAKE COVER UNDER FIRE SCENARIO

The common scenario used during the experiments to evaluate the different behavior approaches was a basic Individual Combatant (IC) behavior. The goal of the scenario was to have the IC react to incoming direct fire by returning fire and/or changing stance to hidden, prone, crouching, or standing. Overall, the scenario contained two ICs. The first IC, a blue IC, was created with callsign A11, at location (0.0, 0.0), in the crouching stance, and with unlimited ammunition. The blue IC was created with the behavior approach under experiment and the Take Cover under Fire order. The second IC, a red IC, was created with callsign B11, at location (0.0, 250.0), in the crouching stance, and with unlimited ammunition. The red IC was created with the

FSM behavior approach and the order to simply fire at any enemy targets sensed.

The scenario was further defined by the goal of the mission. The goal of the mission is embodied in the reward sets established for a scenario. In our experimentation, we defined three different reward sets. Each reward set resulted in a slight variation of the scenario defined above. In the first variant, the main goal was to stay alive and the secondary, less important goal was to kill the enemy IC. In the second variant, the main goal was to kill the enemy and the secondary goal was to stay alive. In the third and final variant, the goals of staying alive and killing the target were equal (balanced). These goals are described by the different numeric reward sets used for each scenario variant (see Figure 1).

Reward / Reward Set	Target Killed	Self Killed
Stay Alive	+10	-100
Kill Enemy	+100	-10
Balanced	+50	-50

Figure 1. Take Cover under Fire Scenario Variant Reward Sets

The overall goal, during experimentation, was to maximize the numeric value of the reward at the end of the scenario. Normally, the determination of rewards would be developed using military doctrine and subject matter expert course of action selection. However, for our scenario variations the rewards were simply defined to allow exploration of the various learning approaches and do not necessarily reflect military doctrine.

SCENARIO SPECIFIC DESIGN

The design and implementation of the scenario was embodied in the development of the Take Cover under Fire behavior for each of the different behavior approaches. Overall, all scenario variant implementations across the behavior approaches have many components in common for general execution. These details can be found in "The Implementation Of Multiple Behavior Approaches using the CGF Test-bed" (Gugel, Pratt, & Smith, 2001). For example, the behavior approaches all utilized the same methods for the order itself as well as to interface with the rest of the infrastructure. Each approach defined the actions and rewards using the scenario definition as a guide. The following sections include the specific design and implementation for each scenario and behavior approach.

Finite State Machine (FSM) Approach

The FSM approach contains a set of states, conditions, and actions. The approach hard-codes the behavior based on military doctrine and subject matter expert input (although our implementation was designed without subject matter input).

For the Take Cover under Fire scenario, the FSM implementations required unique FSMs for the different scenario variants. For the Staying Alive variant the FSM design includes three states: start, Change Stance, and end (see Figure 2). The FSM immediately transitions to the Change Stance State where the stance is changed to Hidden. In this stance, the IC cannot be sensed or fired upon and the IC cannot fire. Once this action is completed, the FSM transitions to the end state where the FSM waits for a new order or additional direction.

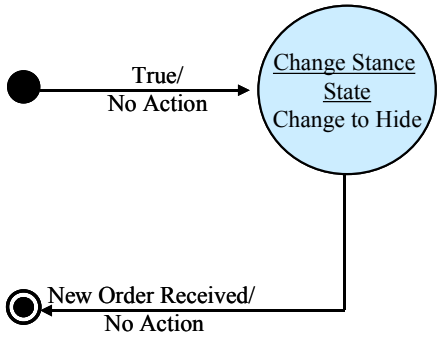

Figure 2. Staying Alive Variant FSM Design

For the Killing the Enemy and Balanced variants, the FSM included four states: start, Change Stance, Return Fire, and end (see Figure 3). The FSM immediately transitions to the Change Stance State where the stance is changed to Prone. Once this action is completed, the FSM transitions to the Return Fire state where the IC fires at the enemy target until a new order is received.

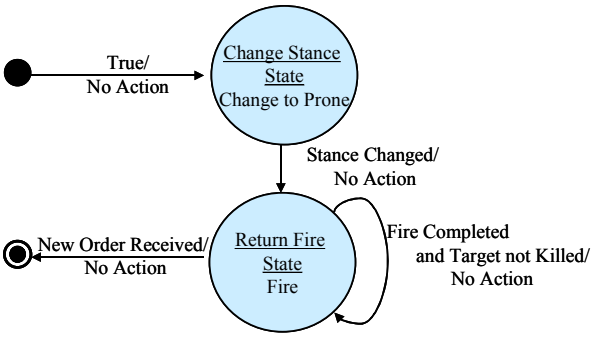

Figure 3. Killing the Enemy & Variant FSM Design

Q-Learning Approach

The Q-Learning approach utilizes Watkin's Q-Learning reinforcement learning algorithm to determine the action to select for each given situation (Watkins, 1986). The Q-Learning implementation includes defining the allowable actions, the states, the action/state mapping, the possible rewards, the reward functions, the Q-Value matrix structure, and Q-Values (Kaelbing and Littman, 1996).

The actual design of the Take Cover under Fire scenario was more complicated than the FSM approach although only one implementation was needed for the different variants. The common design and implementation included the allowable actions, the states, the action/state mapping, and the reward functions. The actions were common to all behavior approaches and included fire, hidden, prone, crouching, and standing. Four states were defined; one state for each stance that IC could be in. The state/action mapping transitioned the IC to the corresponding "stance" state based on the value of the IC's stance in the simulation. Finally, the reward function computed the target killed reward based on the simulation status of the red IC and the self killed reward based on the simulation status of the blue IC at the end of the scenario. The Q-Values matrix was defined by a two dimensional array with states across one axis and actions across the other.

The Q-Learning data that was unique to the Staying Alive variant include the reward values and the actual Q-Values. The reward values used are the self killed and target killed values described above. These reward values are critical to the development of the Q-Values during the learning iterations. The resulting Q-Values for the different scenario variants are described in the experimentation section below.

Evolutionary Approach

The evolutionary approach is based on the principle of evolution and survival of the fittest (Russell & Norvig, 1996). In this approach, a genome is constructed that represents information to determine the sequence of actions that the IC would take in a given situation. For the Take Cover under Fire scenario, fifty genomes (a generation) were created where each genome consisted of 16 genes. Each gene had a value from 0 to 4 representing one of the IC actions. The genomes of a generation were then simulated and rewards were generated. During execution, each genome is iterated through so that each iteration would execute a specific IC action. The genomes were then sorted by the rewards received. The top winning third of genomes was promoted to the next generation. This third was then used as parents for the crossover of genes to create the second third of the generation. The final third was created by randomly selecting one of the genomes in the new generation and randomly changing some of the gene values.

The data that was unique to the Staying Alive variant includes the reward values and the produced generations. These reward values are critical to the development of the genomes during the learning iterations. The resulting "winning" genomes for the different scenario variants are described in the experimentation section below.

Fuzzy Rule-Based Approach

The fuzzy rule-based approach utilizes fuzzy logic to select a course of action from a predetermined set (Pirjanian & Christensen, 1997). The actual actions performed are then tailored as appropriate using preferences posted by losing courses of action. The two courses of action defined for the Take Cover under Fire Scenario include Take Cover and Return Fire (see Figure 4). The selection of the course of action is based on the fuzzy logic equation associated with each course of action (objective node).

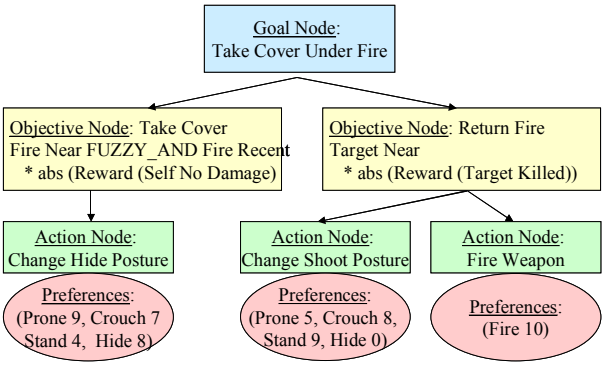

Figure 4. Take Cover under Fire Fuzzy Rule-Based Design

For the scenario, we utilized the absolute values of the rewards to weight which course of action was more "appropriate" for the selected mission. These values were used in association with two different fuzzy functions: near and recent. Both were defined as simple linear values. For near, the distance between the ICs and incoming fire impact location or target location was computed as (500-distance/500) (see Figure 5). Any distance greater than 500 meters was considered not near and thus a value of 0.0 was returned. A similar function was created for Recent (see Figure 6).

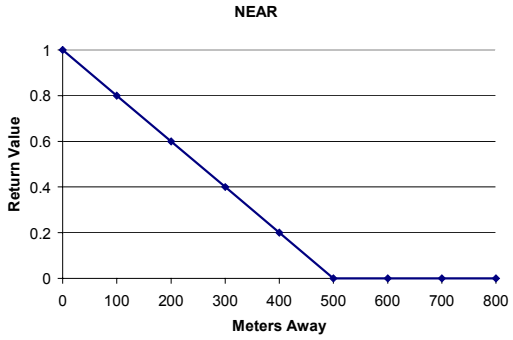

Figure 5. Fuzzy Near Function

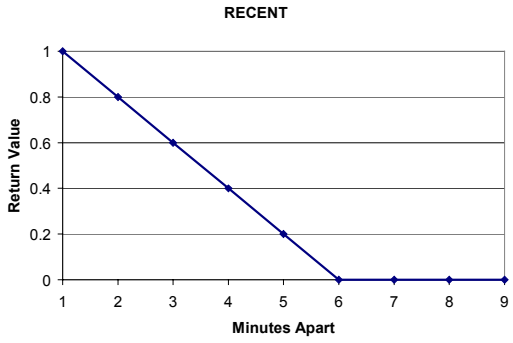

Figure 6. Fuzzy Recent Function

EXPERIMENTATION

Experimentation encompasses several different activities. The first was to modify the CGF Test-bed and the behavior approaches to utilize the rewards for the scenario variant. The required changes amounted to modifying the values of the rewards within the code for the Q-Learning, evolutionary, and fuzzy rule-based approaches. In the future, these rewards should be specified as data to allow changes to be made more easily. The remaining approach, the FSM approach, required the construction of a new FSM to handle the Killing the Enemy and Balanced FSMs.

The second activity was to construct scenario input files for each of the four different behavior approaches. These scenarios specified the data used to create the blue and red ICs including the behavior approach used to control them and a flag noting if the execution was for learning or for test. In the learning mode, the Q-Learning and evolutionary updated the Q-Values and genomes' rewards respectively. In the test mode, no changes were made to the genomes or to the Q-Values to ensure no advantage across the different test runs for the learning approaches.

The third activity was to perform the learning iterations required by the evolutionary and Q-Learning approaches. To run these iterations, the CGF Test-bed was run in an as-fast-as-possible, batch mode using scripts. The total time required for the learning iterations were roughly 1.5 hours for the Q-Learning and 6 hours for the evolutionary approach for each scenario variant. Due to the difference in the computation of the winning course of action, the iteration rates differed. Q-Learning continually modifies a single result embodied in the Q-Values. 200 iterations were performed to align these values. The evolutionary approach, on the other hand, started with 20 genomes in a generation. Fifty generations were then produced according to the algorithm described in the design section above.

The final activity was to run the actual experiments. During the actual experimentation, the CGF Test-bed was run using the GUI mode and with the same scenario files as the learning iterations (but the test flag was set to true to disable future learning). Human interaction was limited to viewing (through the Plan View Display (PVD) and debug statements) and recording of results once the scenario was completed through the GUI entity status window (see Figure 7). Each

approach was tested ten times to allow a sampling of results in the non-repeatable CGF Test-bed.

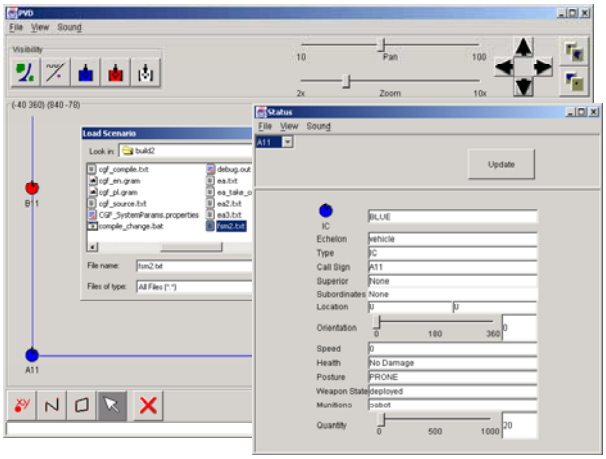

Figure 7. CGF Test-bed User Interface

The following sections outline the results of the experiments for each behavior approach and each scenario variant. In some cases, extra experiments were identified during these planned experiments. These additional results are detailed within the behavior approach and scenario variant where a question was raised that needed further experimentation.

Staying Alive Variant Results

The Staying Alive variant of the Take Cover under Fire scenario weighed the reward for staying alive heavier than killing the enemy. The result that we expected was for the blue IC to change to the hide stance where it could not be sensed or fired upon by the enemy red IC. This hidden stance did not allow the blue IC to fire at the red IC. The results of the experiments did not always match this expectation (see Figure 8).

Approach	FSM	Q-Learning	Evolutionary	Fuzzy
Average # Actions	Infinite	Infinite	7.1	11.9
Minimum # Actions	Infinite	Infinite	3	2
Maximum # Actions	Infinite	Infinite	16	28
% Stayed Alive	100%	100%	10%	0%
% Target Killed	0%	0%	10%	0%
Actions Taken	Hide*	Prone Hide Fire*	0030-4432-4032-3000	Crouch*

Figure 8. Staying Alive Variant Overall Results

Overall, the FSM and Q-Learning approaches both selected actions as we expected (although the Q-Learning did transition to the Prone stance first). The results from the evolutionary and fuzzy rule-based approach were not as successful. The following sub-sections outline the specific behavior approach results.

FSM Approach. For all ten experiments using the FSM approach, the actions were the same since the actions were hard-coded. The continual action was for the IC to change to the hidden stance. The number of actions performed by the blue IC was infinite since from the hidden stance the IC could not be damaged and the scenario time would expire to end the test. In all runs, the blue IC was never killed but also never killed the target.

The Q-Learning Approach. The Q-Learning results are embodied in the Q-Values computed during the learning activity. The actions selected are the highest value for each state (see Figure 9). Since the IC starts in the crouching stance, the next action was selected by picking the action that had the highest value out of all of the crouching state/action pairs. This resulted in the IC changing to the prone stance. Once in the prone stance, the IC selected to go to the hidden stance and then to continually fire. This firing did not occur within the simulation, even though it was selected and the order given, since the blue IC could not sense the red enemy IC to fire at.

State / Action	HIDDEN	PRONE	CROUCHING	STANDING
FIRE	1.1	-0.2	-19.6	-0.5
HIDE	1.1	0.7	-5.5	-8.3
PRONE	-1.2	-21.5	0.5	-15.7
CROUCH	-0.3	-9.0	0.3	-5.8
STAND	-37.6	0.7	-6.6	0.5

Figure 9. Staying Alive Variant Q-Learning Q-Values

The Evolutionary Approach. The evolutionary results are likewise embodied in the learning results, in this case, the genomes. In each generation, any number of genomes can be considered winning. To be a winning genome the simulation result from the last learning iteration had to result in the highest possible reward. For this scenario variant, the largest possible result

was 10 and two different genomes had that result (see Figure 10). The first genome in the list, A0, was used for the experiments to compare against the other behavior approaches. The genome contained no genes to represent the hidden action although the 11TH gene of the second winner A8 did.

IC	Genome
A0	0030-4432-4032-3000
A8	0000-2044-3314-4333

0 = Fire
1 = Hide
2 = Prone
3 = Crouch
4 = Stand

Figure 10. Staying Alive Variant Winning Genomes

To further experiment with the convergence of this approach, the number of generations for this scenario variant was increased to 250 (increased by a factor of 5) to see if the results would yield a better set of winning genomes. The results were better. The blue IC remained alive 40% of the time and the target was killed 30% of the time. In addition, there were five winning genomes in the final generation.

The Fuzzy Approach. The fuzzy results were poor but not because of the fuzzy approach itself. The results were due to the values used to select preferences. This does however show the complexity of developing these values as well as the fuzzy logic equations themselves. For this scenario the value of *target near* was always .5 since the ICs did not move and were 250 meters apart. The value of *fire recent* was above .8 and *fire near* was above .9 after the first incoming round was sensed. This caused the IC to select the Take Cover course of action as expected. The incorrect stance, the crouching stance, was selected in all cases due to the values associated with the preferences for the crouching stance for both courses of action.

The results show that as the rewards are changed in the fuzzy rule-based approach the values associated with the preferences also need to be evaluated and perhaps changed. In addition, as the simulation progresses due to more current situational information, these preferences should change. For the Staying Alive variant, the hidden posture should have been weighted more heavily. The preferences allow the modifiable portions of the actions to be changed so that portions of other courses of action may also be satisfied. For example, if a course of action was selected for an IC to move to a covered position. The IC may also satisfy another goal (for the fireteam to cover the complete sector) by moving to the location that also offers the best position to cover portions of the fireteam's sector not covered by other ICs. In this example, the selection of the covered location was determined both by the winning and losing courses of action.

Killing the Enemy Variant Results

The Killing the Enemy variant of the Take Cover under Fire scenario weighed the reward for killing the enemy heavier than staying alive. The result that we expected was for the blue IC to change to a stance other than hidden, preferably the standing stance where it could continually fire upon by the enemy red IC. The results of the experiments were not as good as the previous variant (Figure 11).

Approach	FSM	Q-Learning	Evolutionary	Fuzzy
Average # Actions	11	8.4	10.1	13.2
Minimum # Actions	4	2	5	2
Maximum # Actions	23	20	16	28
% Stayed Alive	20%	30%	40%	40%
% Target Killed	20%	30%	30%	40%
Actions Taken	Prone Fire*	Fire*	2130-1330-4204-1021	(Crouch Fire)*

Figure 11. Killing the Enemy Variant Overall Results

Overall, the FSM, Q-Learning, and Fuzzy approaches all selected action close to what we expected (although it was interesting that the Q-Learning and Fuzzy approach transitioned into the crouching stance before firing). The fuzzy rule-based approach had the best results with killing the target 40% of the time. The results from the evolutionary approach were not as successful. The following sub-sections outline the specific behavior approach results.

The FSM Approach. For the FSM approach, the red IC won over the blue IC in 8 of the 10 tests. We expected that the blue IC would kill the red IC about 50% of the time since the probabilities for damaging the enemy target were the same for both and both were executing the same behavior. Instead, the blue IC only killed the red IC 20% of the time. For verification, an extra experiment was then run with both IC using the FSM approach and the same "fire" order to ensure that neither had an

advantage. The results were that in 6 of the 10 runs the blue IC was killed and in the other 4 cases the red IC was killed. In general, neither seems to have an advantage over the other. Although part of the difference is a result of the red IC not having to process the new order before starting to fire. Further testing is required to determine the exact cause of the unusual result.

The Q-Learning Approach. The Q-Learning results were a little better than the FSM results. For the Q-Learning approach the IC killed the enemy target 30% of the time. The Q-Values that the IC learned had the IC remain in the initial crouching stance and then simply return fire (see Figure 12). The shortfall from the 50% you would expect (since both IC were performing the same action) is again possibly due to the processing of the order itself.

State / Action	HIDDEN	PRONE	CROUCHING	STANDING
FIRE	76.2	70.2	83.7	74.7
HIDE	76.2	75.9	79.1	90.4
PRONE	73.1	72.0	71.0	73.1
CROUCH	82.4	79.9	77.1	80.0
STAND	74.3	77.5	77.6	72.6

Figure 12. Killing the Enemy Variant Q-Learning Q-Values

Since this learning seemed to center on firing, another run was made increasing the number of iterations to 1000 (an increase of a factor of 5) to see if it altered the result or if it remained the same. The results were surprising (and will require future experimentation). The resulting Q-Values combined with the IC starting in the crouching stance cause the IC to continually change stance and never fire. As a result, the blue IC would never kill the enemy.

The Evolutionary Approach. The evolutionary results provided several winning genomes of which the A1 genome was selected (see Figure 13). The genome actions included changing to various different stances and firing. The other winning genomes were interesting because they start with almost the same actions: fire, fire, prone, fire, and hide. To explore the results of these genomes, genome A10 was also selected to run experiments on. The experiment resulted in the blue IC staying alive 30% of the time and the red IC getting killed 20% of the time which was actually worse than results outlined above (see Figure 11).

IC	Genome
A1	2130-1330-4204-1021
A2	0120-1030-4204-1231
A3	0020-1331-4204-1201
A5	0020-4430-4032-3000
A10	0020-1033-4000-0204

0 = Fire
1 = Hide
2 = Prone
3 = Crouch
4 = Stand

Figure 13. Killing the Enemy Variant Winning Genomes

The Fuzzy Approach. The fuzzy results were the best for this scenario variant. The fuzzy design resulted in the actions of changing stance to fire and then firing. The values of the wining and losing course of action caused the crouching stance to be selected. In our implementation, the IC would not change stance if already in that stance and would instead perform the next action. In this manner, although the results show that the IC switched between the crouch and fire actions, the crouch actions had no effect on the simulation.

Balanced Variant Results

The Balanced variant of the Take Cover under Fire scenario weighed the reward for killing the enemy and the reward for staying alive the same (although one was a negative reward and the other a positive reward). We expected the blue IC to change to a stance other than hidden (preferably the prone stance to reduce being killed) where it could continually fire upon by the enemy. The actual actions selected by the behavior approaches were close to this expected action (see Figure 14).

Overall, the evolutionary approach yielded the best results. The actions involved firing and changing stance. The FSM and Q-Learning approaches selected the same actions: to change stance to prone but then to fire. The fuzzy approach, in most runs, would crouch and return fire to achieve the return fire course of action but then upon receiving fire would change to the crouch stance and remain there. This problem was described in the first scenario variant where the preferences did not consider the change in reward values and did not change over time to reflect the long-term effects of events within the simulation. The following sub-sections outline the specific behavior approach results.

Approach	FSM	Q-Learning	Evolutionary	Fuzzy
Average # Actions	11	7.8	7.5	10.8
Minimum # Actions	4	4	3	3
Maximum # Actions	23	14	16	33
% Stayed Alive	20%	20%	30%	10%
% Target Killed	20%	20%	30%	10%
Actions Taken	Prone Fire *	Prone Fire*	0440-0130-4014-0012	Crouch Fire Crouch*

Figure 14. Balanced Variant Overall Results

The FSM Approach. The FSM results were used from the set of runs from the previous scenario variant as the same behavior satisfied the scenario variant. Refer to that section for details.

The Q-Learning Approach. The Q-Learning results resulted in the selection of actions as had been expected (see Figure 15). The overall Q-Values show that if the IC is in the hidden stance then it stays there so that it cannot be killed. On the other hand, if the IC is in one of the other stances then it transitions into the prone stance and, once there, starts firing upon the enemy IC. This result works to satisfy either, but not both of, the two goals: staying alive and killing the enemy.

State / Action	HIDDEN	PRONE	CROUCHING	STANDING
FIRE	25.3	25.2	18.2	24.7
HIDE	26.3	19.6	22.5	26.6
PRONE	0.8	18.0	31.3	17.5
CROUCH	23.1	18.9	10.8	28.3
STAND	17.9	24.1	24.5	22.3

Figure 15. Balanced Variant Q-Learning Q-Values

The Evolutionary Approach. The evolutionary results produced two winning genomes (see Figure 16). For the experiments, as before, the first genome was selected for testing, A3. The genomes were more than likely offspring of at least one of the same parents since they have 13 genes with the same value out of the 16 genes. Due to this close relationship, these genomes might both be mutations of winning genomes from the previous learning iteration. In either case, the A3 genome provided the best results for this scenario variant.

IC	Genome
A3	0440-0130-4014-0012
A6	0400-0130-4014-0103

0 = Fire
1 = Hide
2 = Prone
3 = Crouch
4 = Stand

Figure 16. Balanced Variant Winning Genomes

The Fuzzy Approach. The fuzzy results displayed problems similar to the Staying Alive variant. Since both courses of action had the same reward factor of 50. The selection of the course of action was dependent on the fuzzy logic values themselves. Once the enemy red IC started firing, the Take Cover course of action was ranked the highest and the couch stance selected. In production CGF system, subject matter experts, who could foresee such conflicts and help to resolve them early during design, would review these fuzzy logic equations and preferences to ensure they correspond to doctrine.

OVERALL RESULTS/LESONS LEARNED

Overall the results showed that in various instances different approaches proved the best. In fact, each one of the behavior approaches ranked the highest (or tied for the position) across the three scenario variants. Throughout the experimentation, various other observations were made as to the applicability of these approaches for modeling and simulation. Five of these observations follow.

First, changing the rewards for the Q-Learning, evolutionary, and fuzzy rule-based approach was straightforward. The change required an update to a constant. (In future implementations the reward should be stored in a data file.) The impact was larger to the FSM approach, however. It required that a new FSM be designed, developed, and tested. The Q-Learning and evolutionary did have the drawback that the learning had to be performed whenever the rewards changed.

Second, directly comparing the different learning methods was difficult. Determining if one had the advantage over the other, due to the number of learning iterations performed, was impracticable to gauge because of the different principles of the approaches. In addition, determining how many

learning iterations to perform to ensure a reasonable result was also hard to determine. In our experiments, the numbers were chosen based on initial tests with the Q-Learning method and an attempt to allow equal time for learning by the evolutionary approach.

Third, the determination of when to select the next action could be enhanced across all implementations. Work was done before the experiments to, at a minimum, allow the IC to fire as many rounds as possible when the fire action was specified. Usually, one to three rounds were fired. This helped to even the playing field between the red and blue IC while still controlling the blue IC with the behavior approach. An extension could also be added to allow actions to occur at the same time but the behavior approach designs would have to be changed to allow this possibility.

Fourth, several possible enhancements were noted for future testing of the learning approaches. For the Q-Learning approach, the learning rate could be altered as the learning progresses to help yield a better result since the Q-Values are closer to the optimal solution as the learning progresses. For the evolutionary approach, error checking should be performed to ensure that no genes specifies an invalid action. In addition, the code that produces the generations could be modified to only allow non-losing genomes to be transferred to the new generation. This includes limiting the mating to only winning genomes. Finally, the evolutionary approach could be modified to allow multiple runs for each genome within a generation to yield a more statistical result for the rewards assigned to each genome during the learning phase. This enhancement would however increase the learning time needed.

Finally, additional scenarios and behaviors should be created to explore behaviors that require a series of steps to complete an action such as moving past an enemy target while optionally returning fire. These more complex scenarios will further enhance our understanding of the advantages and disadvantages of each approach.

Through our experiments and ongoing work using these behavior approaches we have noted the advantages and disadvantages of the different behavior approaches.

The FSM approach is straightforward to implement and efficient. Through existing programs such as ModSAF and CCTT, the FSMs have been proven scaleable and maintainable when common primitives are utilized for common functions. The addition of new behaviors in the FSM approach is straightforward and includes mainly designing and implementing the new behavior. The insertion of new simulated object attributes such as adding an IC's fatigue to an existing behavior is difficult since every FSM must be reviewed for applicability. Each applicable FSM must then be modified and tested. The FSM approach does however prove to be suitable for the strict encoding of doctrine.

The Q-Learning approach has proven through these experiments to be maintainable and easy to implement. Through the design for the scenario, it was easy to see that the approach did not scale well. The states required to represent an IC in a more complex scenario would increase drastically. For example, if fatigue were added to the existing scenario design then the state space would double simply to represent the presence and absence of fatigue while the IC is in each of the stances. As can be seen, the addition of a new simulated object attributes is not as hard as the FSM approach but does require the regeneration of the learned Q-Values. Finally, it is the exploratory nature of the approach in selecting actions that provides the ability to see what an ideal action may be in the investigation of new doctrine and force structures. It is not, however, ideal for the strict encoding of doctrine as seen by the varying actions selected by the learning methods.

Like the Q-Learning approach the evolutionary approach is also ideal for exploring and the learning can result in non-doctrinal behaviors. Although perhaps not suited for encoding exact military doctrine, learning approaches may be helpful in exploring new doctrine based on changing weapons, changing battlefield environments, and changing enemy tactics. Overall, the approach is easy to implement. The definition of the genome representation is the hardest part of designing the evolutionary approach for a specific scenario. For our implementation, it was a predefined number of actions. The insertion of new behaviors is easy and the addition of new simulated object attributes is simple but it requires the regeneration of the "winning" genome through learning iterations.

The fuzzy approach is a hard-coded approach as the FSM approach is. The approach provides a more realistic decision making ability than the strict FSMs both through the use of fuzzy logic and

also through the use and application of preferences. The two together in a hybrid implementation with the fuzzy logic making the decisions and the FSM implementing the basic behaviors provides the efficiency and flexibility of both approaches. On its own, the fuzzy approach is straightforward to implement, is scaleable, and is maintainable. It also allows the introduction of new simulated object attributes into the decision making with little change. Finally, this approach allows for the strict encoding of doctrine.

CONCLUSIONS

Throughout our efforts we found that in all cases (although especially for the learning algorithms) the results are only as good as the data used within the simulation. If the data used by the simulation is altered (for example to change data from classified to unclassified) then the resulting actions learned may not be the most effective in real life. As far as selecting a behavior representation for use, it will depend on the behavior being executed including the amount of decision making it requires, the nature of the behavior (doctrinal and non-doctrinal) as well as the behavior usage (training or research). In addition, hybrids of these solutions may be found to provide the best approaches. However, it is clear that future CGF simulations will need to provide the ability to encode differing approaches to handle these different cases and to also allow more research in a complete CGF system.

REFERENCES

Gugel, S., Pratt, D., & Smith, R. (2001). The Implementation of Multiple Behavior Approaches in the CGF Test-bed. Proceedings of the 10th Computer Generated Forces and Behavior Representation Conference, Norfolk, VA. May 15-17, 255-266.

Pew, R., & Mavor, A. (Ed.) (1998). Modeling and Human Organizational Behavior. Washington, D.C.: National Academy Press.

Courtemanche, A. (2000). The Incorporation of Validated Combat Models into a Discrete Event Simulation. Proceedings of the 9th Computer Generate Forces and Behavior Representation Conference, Orlando, FL, May 16-18.

Watkins, C. J. C. H. (1986). Learning From Delayed Rewards. Thesis. University of Cambridge, England.

Kaelbing, Leslie Pack and Littman, Michael L. (1996). Reinforcement Learning: A Survey. Journal of Artificial Intelligence Research 4 (1996) pp. 237-285.

Russell, Stuart and Norvig, Peter (1995). Artificial Intelligence: A Modern Approach. New Jersey: Prentice Hall.

Pirjanian, Paolo and Christensen, Henrik (1997). Behavior Coordination Using Multiple-Objective Decision Making. SPIE Conference on Intelligent Systems and Advanced Manufacturing, Pittsburgh, Pennsylvania, 14-17 October.