

AFRL-RI-RS-TR-2007-223
Final Technical Report
October 2007



AN AGENT POPULATED TESTBED FOR C2 EXPERIMENTATION

Stottler Henke Associates, Inc.

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Air Force Research Laboratory Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-RI-RS-TR-2007-223 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/

/s/

ALEXANDER SARNACKI
Work Unit Manager

ROBERT S. MCHALE, Deputy Chief
Information Systems Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE				<i>Form Approved</i> OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.					
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) OCT 2007		2. REPORT TYPE Final		3. DATES COVERED (From - To) Jun 05 – Jun 07	
4. TITLE AND SUBTITLE AN AGENT POPULATED TESTBED FOR C2 EXPERIMENTATION				5a. CONTRACT NUMBER FA8750-05-C-0208	
				5b. GRANT NUMBER 	
				5c. PROGRAM ELEMENT NUMBER 62702F	
6. AUTHOR(S) Alex Davis, Dan Fu and John Rushing				5d. PROJECT NUMBER 558B	
				5e. TASK NUMBER ST	
				5f. WORK UNIT NUMBER OT	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Stottler Henke Associates, Inc. 951 Mariner's Island Blvd, Ste 360 San Mateo CA 94404				8. PERFORMING ORGANIZATION REPORT NUMBER 	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFRL/RISB 525 Brooks Rd Rome NY 13441-4505				10. SPONSOR/MONITOR'S ACRONYM(S) 	
				11. SPONSORING/MONITORING AGENCY REPORT NUMBER AFRL-RI-RS-TR-2007-223	
12. DISTRIBUTION AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# WPAFB-07-0071					
13. SUPPLEMENTARY NOTES 					
14. ABSTRACT Recent concepts in the field of command and control (C2), such as Power to the Edge and Network-Centric Warfare, have indicated the need for a testbed for experimentation. We describe a gaming testbed, populated by realistic synthetic agents for modeling the complex human interactions comprising C2 structures, and for exploring the effectiveness of C2 concepts in a variety of tactical circumstances.					
15. SUBJECT TERMS Serious Games, Artificial Intelligence, Genetic Algorithms, Data Mining, C2, Command and Control, Wargame					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UL	18. NUMBER OF PAGES 22	19a. NAME OF RESPONSIBLE PERSON Alexander Sarnacki
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (Include area code) N/A

ABSTRACT

Recent concepts in the field of command and control (C2), such as Power to the Edge and Network-Centric Warfare, have indicated the need for a testbed for experimentation. We describe a gaming testbed, populated by realistic synthetic agents, for modeling the complex human interactions comprising C2 structures, and for exploring the effectiveness of C2 concepts in a variety of tactical circumstances.

A testbed for experimentation with C2 concepts must be capable of reproducing the complex, rapid, dynamic, and unpredictable unfolding of events in battle. Scenarios can be carefully limited, and realism diluted to a practical level, but enough of the complex interactions of individual and collective behavior must be preserved to capture the essentially human nature of C2. Experimentation involves the correlation of tactical events and outcomes with a host of variable parameters, such as C2 structure, communication patterns and reliability, and implementation of commander's intent.

This report describes the details of our testbed and experiment design and implementation, in addition to findings and lessons learned. Our approach included the integration and extension of three existing platforms: Counter-Strike, a multiplayer first-person tactical shooter; SimBionic, a visual behavior authoring and execution engine developed at Stottler Henke; and ADaM, a data mining tool suite developed at the University of Alabama, Huntsville. The testbed produced by this integration allows for the planning and execution of missions by human or synthetic agents, including specifications of different C2 structures, and the mining of experimental data for the effectiveness of the C2 concepts. We discuss the applicability of our findings to other echelons, as well as its contribution to the evaluation of new C2 concepts.

TABLE OF CONTENTS

1	INTRODUCTION	1
2	TESTBED DESCRIPTION.....	2
2.1	The Counter-Strike Game	3
2.1.1	Counter-Strike AI	3
2.1.2	Counter-Strike Scenario	4
2.2	C2 Configuration.....	4
2.3	ADaM Data Mining.....	4
2.3.1	Data Mining Analysis for C2	4
2.3.2	Genetic Algorithms for Evolving C2.....	5
2.4	Visualization	5
3	ANALYSIS.....	7
4	LESSONS.....	11
4.1	Fitness as Kill Ratio	11
4.2	3-D Game Mechanics	11
4.3	Real-time Experimentation	12
4.4	Scenario Scope.....	12
5	CONCLUSION.....	14
6	REFERENCES	15

LIST OF FIGURES

Figure 1.	Testbed overview.....	2
Figure 2.	Log playback tool.....	6

LIST OF TABLES

Table 1.	First phase experimental results.....	7
Table 2.	Sample bitstring encoding for genetic algorithm.....	8

PREFACE

The purpose of this project was to establish the feasibility of a low-cost testbed for the exploration and analysis of C2 concepts, and to identify the necessary technologies for its implementation.

This included the selection of a COTS game-based 3-D software platform, integration of that platform with authorable agent behaviors, development of such behaviors to allow agents to operate in a variety of C2 configurations, instrumentation of the platform for collection of game data in real time, and provision of data mining analysis tools which could process that data and produce human-understandable results.

SUMMARY

A testbed prototype was constructed, linking Counter-Strike (CS), an inexpensive COTS 3-D game simulating counterterrorist small-unit combat, with a variety of tools developed by Stottler Henke and Univ. of Alabama Huntsville. These tools included a visual editor for configuration of an experiment or series of experiments; instrumentation of the CS engine to record player position and activity data in real time; a visualization tool for playback of a single game; data mining tools for the analysis of instrumentation data; and various software to interconnect and automate the use of this group of tools as a testbed.

1 Introduction

Recent and emerging theories in the field of command and control (C2), such as *Power to the Edge* (Alberts et al, 2003) and *Network-Centric Warfare* (Alberts et al, 2002), have afforded compelling ideas of how teams might be organized to maximize their effectiveness. For empirical purposes, there exists a need for testbeds which can serve as experimental environments for new ideas. A testbed for experimentation with C2 concepts must be capable of reproducing the complex, rapid, dynamic, and unpredictable unfolding of events in battle. Scenarios can be carefully limited, and realism diluted to a practical level, but enough of the complex interactions of individual and collective behavior must be preserved to capture the essentially human nature of C2.

Experimentation involves the correlation of tactical events and outcomes with a host of variable parameters, such as C2 structure, communication patterns, and reliability.

At the same time as C2 theories gain currency, there has been a timely interest in leveraging game-based technology platforms, specifically for training and simulation purposes. These platforms feature already-completed engines, tools, and content which present low cost, low risk ways to build useful serious games. In this report we describe our efforts to produce a game-based experimental testbed meant to enable researchers to:

- **Configure C2 structures:** Specify how command and communication happens between individuals.
- **Adjust performance parameters:** Change performance criteria of communication and command.
- **Visualize logged data:** Play back logged data.
- **Evolve behavior:** Genetic Algorithms (GA) vary structures and performance parameters to derive new behavior.
- **Perform data mining:** Run standard data mining algorithms on logged data to find patterns of interest.

Our approach includes the integration and extension of three existing platforms: a multiplayer game, AI authoring tool, and data mining tool suite. The resulting testbed produced by this integration allows for the execution of missions by human or synthetic agents, including specifications of different C2 structures, automated evolution and evaluation of those structures, and the mining of experimental data for the effectiveness of C2 concepts.

In this report we describe the details of our testbed, experiment design and implementation, in addition to findings and lessons learned.

2 Testbed Description

Our testbed comprises three main components:

- **Counter-Strike game:** a 3-D tactical shooting game pitting terrorist versus counter-terrorist;
- **SimBionic AI authoring tool:** a visual behavior authoring and execution engine developed at Stottler Henke (Fu and Houlette, 2002); and
- **ADaM data mining tool suite:** machine learning software developed at the University of Alabama in Huntsville (Rushing et al, 2005).

A **C2 configuration** and **log playback tool** complete our testbed. Figure 1 shows the testbed's organization. It operates in one of two modes. In the first, the "configuration tools" are for a researcher who specifies a C2 configuration and performance parameters for an experiment. The Counter-Strike game server starts and begins a round. The SimBionic AI tool uses the data to instantiate non-player characters (NPC's) that will conform to the experiment's specifications. Human participants may take part in these experiments. After the round is completed, the user may view the logged data through a playback tool, or invoke the data mining toolkit to search for interesting data.

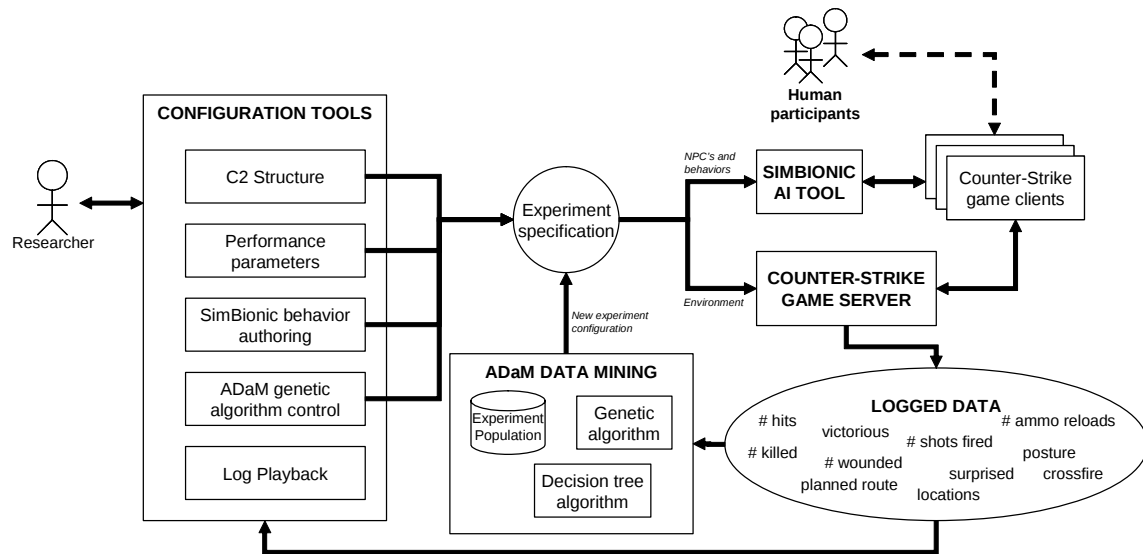


Figure 1. Testbed overview.

In the second mode, the testbed will automatically experiment with various configurations to discover effective C2 structures. Here, the testbed uses a genetic algorithm to explore the space of possible structures, discovering, combining, and testing successful C2 structures and evaluating their efficacy. In the rest of this section, we describe the components in more detail before proceeding to analysis in the next section.

2.1 The Counter-Strike Game

Counter-Strike is a 3-D multiplayer first person shooter game based on the Half-Life game. While this game hosts up to thirty-two human players, there exist alternate ways to insert NPC's into the game. Their participation in the game is similar to a human player. An important distinction of Counter-Strike is its encouragement of team-based gameplay. Individual players are rewarded only if their respective team wins. This implies that successful members coordinate and support each other as opposed to "death matches" where everyone competes against each other.

Our approach using this game is from the bottom up, with a low level, tactical game that can provide interesting results in the evaluation of C2 concepts, and form a basis for C2 analysis techniques that potentially scale to higher levels of command. Squad-based use of CS gives three types of actors in the chain of command:

1. **Soldiers in the field**, having direct interaction with the mission environment producing immediate effects, embodying a fast-paced, stressful environment for mission execution and leadership in the field.
2. **Intermediate C2 staff** that must coordinate different units and levels of command. They may be present in the field, in a staging area, or offsite.
3. **Command leadership** that must enact intent, with varying levels of intervention into mission execution.

Scenarios in Counter-Strike involve up to a platoon-sized contingent in tactical urban operations, such as clearing buildings or rescuing hostages held by terrorists. Communications are all-channel radio broadcast of simple commands and information, and control is left up to the players in the field. We augmented this system with explicit C2 mechanisms, allowing players on and off site to monitor and adjust the close battle, as well as leadership and functional roles that transform the free-for-all environment of commercial Counter-Strike into optional elements such as fire teams and squads fulfilling assault and support roles.

This allows for some low-level versions of C2 elements, such as unit coordination and logistics, while preserving fidelity and individuality of behavior at both command and execution levels.

2.1.1 Counter-Strike AI

NPC's are capable of a variety of basic behaviors, such as following specified paths or other agents, searching, and engaging observed enemies (Fu, Houlette, Jensen, & Bascara, 2003). They also are capable of more complex behaviors such as taking cover on contact, calling for backup, responding to calls, and coordinating with another team of agents to prepare and time a synchronized attack. These behaviors are configured through the testbed into C2 schemes; for instance, a command hierarchy is formed by establishing each agent's policies for whom to follow (or lead), with whom to communicate, and how to react to various events.

2.1.2 Counter-Strike Scenario

The baseline scenario within which these agents operate includes two opposing teams, CT (counter-terrorist) and T (terrorist). They start at opposite ends of the map and attempt to destroy the opposing team, at which point the game ends. Our analyses tended to stage the T in a defensive position that the CT would have to assault, but the testbed supports a variety of other scenarios, such as the T sending some members to sniper locations, or both sides actively searching for the enemy.

2.2 C2 Configuration

The configuration editor is the starting point for the end-to-end analysis cycle. This editor allows the experimenter to set up each team with an arbitrary command hierarchy expressed as trees, and with arbitrary communications scheme expressed as nets of common communication between agents. It also allows the configuration of a variety of variables that affect gameplay and decision making.

Two of the variables correspond to command: **promotion delay**, which is the time that must pass before a fallen leader is replaced, and **backup contingent**, which is the number of units sent in response to a backup call. These parameters are intended to implement a portion of C2 policy, as an abstraction of the effects of decisions made by an on-scene or off-scene commander, or the by the teams themselves.

Two other variables affect communications: **drop probability** and **message delay**. These variables simulate variance in reliability of communications, and in their time of travel along different avenues (for instance, via a third off-scene party).

Other variables are also configurable, chiefly as parameters to agent decision-making behavior, such as whether a team defends, searches for targets, or attacks along planned routes (which are also configurable), and whether agents attempt to cover and synchronize engagements upon contact with the enemy. These variables are provided in order to implement particular policies, plans, and scenarios within the baseline scenario, and to guide the action into particular points of interest, such as the moment of synchronized assault on a defensive position (Stottler, Lackey, & Kirby, 2004).

2.3 ADaM Data Mining

ADaM (Algorithm Development and Mining) is a freely available data mining toolkit designed for use with scientific data. The ADaM toolkit provides a suite of tools for each of the basic data mining processes, including classification, clustering, association rule mining, and preprocessing. The toolkit is packaged as a series of independent components. Each component can be used either as a standalone executable, or from within the Python scripting language via a wrapper. As mentioned earlier, ADaM's role within the testbed is to (1) data mine for patterns, and (2) evolve more effective C2 structures.

2.3.1 Data Mining Analysis for C2

We use decision trees to determine which C2 parameters contribute most to victory. To accomplish this, the logged data serves as input to ADaM, which then produces a decision tree that can be used to predict outcomes. Decision trees select attributes based

on relevance to the category of interest (in this case, victory conditions) using information gain.

2.3.2 Genetic Algorithms for Evolving C2

Genetic algorithms (GA) are general purpose search algorithms that are used to solve difficult optimization problems. Genetic algorithms work by generating potential solutions to the problem of interest in some problem specific search space. Typically, solutions are described by bit strings. The genetic algorithm randomly generates a population of bit strings and evaluates them using an objective function. It then generates a new population of bit strings by recombining the ones from its existing population, with the higher rated strings contributing more to the new population.

Genetic algorithms are used to optimize C2 parameters with respect to a particular scenario. In order to accomplish this, it is necessary to represent the relevant parameters as a bit string for the GA. This is generally done by quantizing each parameter within a fixed range. Assigning more bits to a parameter expands the search space and allows for finer tuning. Given a coding scheme, the bit string is evaluated by running the game engine many times with the corresponding AI settings and deriving statistics from the results.

2.4 Visualization

The testbed is designed to lend insight into the operation of lifelike missions, and so visualization of individual experimental runs is important for understanding the reasons and implications of any given game outcome. Two tools were supplied: the game itself, where an experimenter could fly around in the game space and observe activities, or even participate in them, and a tracking tool which renders the game in a single 2-D view.

For in-game observation, agent and terrain markup is supplied to annotate behaviors. The map shows waypoint locations of various types (such as transit and sniping points), and the agents show their navigating, following, engaging, and communications activities through lines connecting them to waypoints and other agents. This view can only be used while the game is in progress.

The agent tracker, shown in Figure 2, shows similar information, but reads a log file and can be rewound and replayed. It shows a top-down view of all agents on the map (red or blue dots) and their direction of sight (line from dot), and marks events such as deaths ("X"), and engagements connecting shooters to targets (arrow to target).

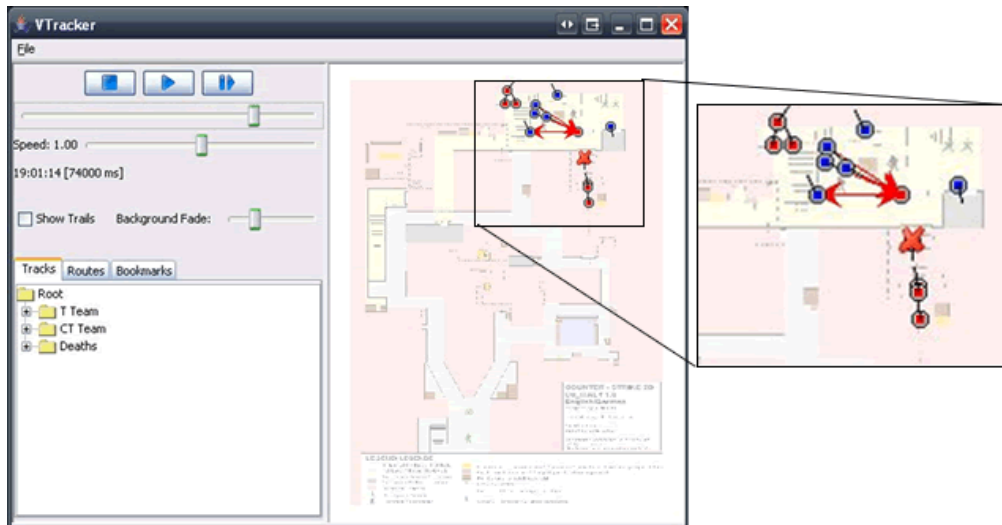


Figure 2. Log playback tool.

3 Analysis

The testbed can be configured to run a predetermined number of times, for a given fixed configuration, or by automatically incrementing variables over specified ranges. Other routines are provided for performing analyses on the logs of these runs, which record moment-to-moment events and agent activities, such as moving, following, pointing, targeting, and engaging. At the end of the process, the testbed produced human-readable results from analysis products. We performed two analyses in order to exemplify findings derivable from the C2 testbed.

The first phase of analysis pitted counter-terrorist (CT) against terrorist (T), each using one of three organizational schemes: hierarchy, hive, and independent. The hierarchy consists of a two-tiered tree, with two teams, and one additional leader whose team was made up of those teams' leaders. The hive is a single-tier hierarchy, with one leader leading all other players on a side. The independent scheme included no teams, each players deciding and acting individually. Along with the nine combinations of those organization schemes, T's were configured to either defend or go on offense, searching the map for targets. The number of units responding to a backup call was also varied.

The full set of possible configurations was run in the testbed, and ranked according to kill ratio, the ratio of CT dead to T. This ranking confirmed expectations regarding game mechanics, such as the fact that shotguns were better one-on-one and at short range, while rifles were superior at long range and in groups; and regarding light infantry engagements in general, such as the superiority of defense to offense. C2 configurations tended to succeed when they favored the appropriate tactic for weapon choice, such as hierarchical search teams, which tended to stay in groups, bearing rifles.

Table 1. First phase experimental results.

CT Structure	T Structure	T Stay	CT Wins	T Wins	Draws
Independent	Independent	No	46	62	0
Independent	Hive	No	43	65	0
Independent	Hierarchical	No	43	65	0
Hive	Independent	No	52	56	0
Hive	Hive	No	42	66	0
Hive	Hierarchical	No	44	64	0
Hierarchical	Independent	No	47	61	0
Hierarchical	Hive	No	42	66	0
Hierarchical	Hierarchical	No	49	59	0
Independent	Independent	Yes	13	90	5
Independent	Hive	Yes	37	71	0
Independent	Hierarchical	Yes	41	66	1
Hive	Independent	Yes	11	95	2
Hive	Hive	Yes	21	87	0
Hive	Hierarchical	Yes	28	78	2
Hierarchical	Independent	Yes	9	95	4
Hierarchical	Hive	Yes	26	80	2
Hierarchical	Hierarchical	Yes	28	78	2

The second phase of analysis focused on decision making by the CT team. It varied weapon choice, planned assault vs. target search, whether to engage targets of opportunity, whether to synchronize assault, and the sizes of the two teams. The hierarchical organization was fixed, as was the T team's defensive posture.

This set of variables was encoded in a bit string and used for the objective function of a genetic algorithm optimizer, using kill ratio as a measure of fitness. We ran two generations of the algorithm, generating ten results for each of 78 bit strings. The results showed the superiority of the rifle to the shotgun (here, engagements tended to initiate at longer range), of large groups to small, and of a search approach to the planned assault. The latter result was the most prominent. Table 2 shows sample bit strings, their composition, and scores.

- SHOT: Whether shotguns were used. A one means yes.
- PLAN: Whether there is a planned assault using two groups. A one means yes.
- ENGAGE: Shows the settings for seven players on a team. A one means that the player engages targets of opportunity on their way to a destination. If zero, the player proceeds to a planned destination.
- SYNC: Whether to synchronize an attack. One equals yes.
- NUM_A: The size of the first group. This only applies if PLAN equals one; i.e., there are two groups on planned routes. The remainder is in the other group. If PLAN is zero, this number has no significance.
- SCORE: The fitness value associated. It indicates a numerical correlation between the bit string and the team winning.

Table 2: Sample bitstring encoding for genetic algorithm.

BIT STRING	SHOT	PLAN	ENGAGE	SYNC	NUM A	NUM S	SCORE
0001110000010001	0	0	0111000	0	2	1	99
0001100000010001	0	0	0110000	0	2	1	74
0000011110110011	0	0	0001111	0	6	3	71
0010110001001001	0	0	1011000	1	1	1	61
1011001000110111	1	0	1100100	0	6	7	61
1001100000010001	1	0	0110000	0	2	1	58
0000100111010000	0	0	0010011	1	2	1	57
0101111001111110	0	1	0111100	1	6	6	55
1010110001001001	1	0	1011000	1	1	1	54
0001100010010111	0	0	0110001	0	2	7	53
0000111111110001	0	0	0011111	1	6	1	47
1010111101010001	1	0	1011110	1	2	1	45
0011001000110111	0	0	1100100	0	6	7	45
0011101110110111	0	0	1110111	0	6	7	45

BIT STRING	SHOT	PLAN	ENGAGE	SYNC	NUM A	NUM S	SCORE
1001010111110101	1	0	0101011	1	6	5	40
0011110110010100	0	0	1111011	0	2	4	35
0000111111110010	0	0	0011111	1	6	2	30
0100100001111100	0	1	0010000	1	6	4	28
1011001100110010	1	0	1100110	0	6	2	28
0011100110101111	0	0	1110011	0	5	7	27
0010011111100000	0	0	1001111	1	4	1	25
0010100001111011	0	0	1010000	1	6	3	24
0001010010010111	0	0	0101001	0	2	7	22
1100111100110010	1	1	0011110	0	6	2	22
1010110000110111	1	0	1011000	0	6	7	21
0111000101110111	0	1	1100010	1	6	7	19
0100010111110100	0	1	0001011	1	6	4	18
0000111111110110	0	0	0011111	1	6	6	18
0100010111110011	0	1	0001011	1	6	3	16
0001100101010010	0	0	0110010	1	2	2	15
0011101110000000	0	0	1110111	0	1	1	14
0100011001010101	0	1	0001100	1	2	5	14
1100111101110110	1	1	0011110	1	6	6	13
0010010101000010	0	0	1001010	1	1	2	13
0011111110110011	0	0	1111111	0	6	3	12
1100101010000010	1	1	0010101	0	1	2	8
1010000110010111	1	0	1000011	0	2	7	7
1100000010101100	1	1	0000001	0	5	4	7
0011111111110001	0	0	1111111	1	6	1	6
0000000101110111	0	0	0000010	1	6	7	5
0001110000111011	0	0	0111000	0	6	3	5
0001001011101101	0	0	0100101	1	5	5	4
0110100100000111	0	1	1010010	0	1	7	4
0011011101100110	0	0	1101110	1	4	6	3
1010111101100010	1	0	1011110	1	4	2	3
1010110001110010	1	0	1011000	1	6	2	3
0000011110010111	0	0	0001111	0	2	7	2
0011110101110111	0	0	1111010	1	6	7	1
1010111101000110	1	0	1011110	1	1	6	-2
1010111101010000	1	0	1011110	1	2	1	-2
1111010110110001	1	1	1101011	0	6	1	-3
0110100100001100	0	1	1010010	0	1	4	-5
0111101100110110	0	1	1110110	0	6	6	-6
0000101001101010	0	0	0010100	1	5	2	-9
0001100000110111	0	0	0110000	0	6	7	-9
0100010010000000	0	1	0001001	0	1	1	-10
1000001100101011	1	0	0000110	0	5	3	-10

BIT STRING	SHOT	PLAN	ENGAGE	SYNC	NUM A	NUM S	SCORE
1011001010100000	1	0	1100101	0	4	1	-11
1100011011110000	1	1	0001101	1	6	1	-11
1010110000010001	1	0	1011000	0	2	1	-13
1111100110001011	1	1	1110011	0	1	3	-14
1111100110101111	1	1	1110011	0	5	7	-15
0111110010110010	0	1	1111001	0	6	2	-19
0011111001111110	0	0	1111100	1	6	6	-25
0111100110001111	0	1	1110011	0	1	7	-25
0110000111010101	0	1	1000011	1	2	5	-27
0100110100001001	0	1	0011010	0	1	1	-27
0101110010110000	0	1	0111001	0	6	1	-28
1101110110101011	1	1	0111011	0	5	3	-32
1111010100110001	1	1	1101010	0	6	1	-33
1100111101010001	1	1	0011110	1	2	1	-33
0000101011001001	0	0	0010101	1	1	1	-37
1100101000011110	1	1	0010100	0	3	6	-41
0111001001001000	0	1	1100100	1	1	1	-45
1111101101010001	1	1	1110110	1	2	1	-45
1110110001100000	1	1	1011000	1	4	1	-47
0101001001000010	0	1	0100100	1	1	2	-62
1111111101100010	1	1	1111110	1	4	2	-74

4 Lessons

In the following, we highlight some of the difficulties and shortcomings of this testbed approach, and indicate potential further work.

4.1 *Fitness as Kill Ratio*

The primary reason that our preliminary analysis did not produce any novel results is that kill ratio is not an ideal measure of fitness for a C2 configuration in this game. Because the game mechanics are not deterministic, and the agent behaviors are not robust enough to handle every kind of contingency, final outcome appears often to derive from other factors than those being studied.

There are, however, other *intermediary analysis products* that can serve better. For synchronization, the testbed could be used to measure the success of the tactic by examining the achieved surprise, as a tally of enemies engaged without seeing their attackers, and crossfire, as a tally of enemies engaged from multiple directions. Synchronized attack in this scenario is designed to maximize these measures. While a great number of experimental runs might be necessary to observe the effects of these achievements on final outcome, the experimenter could optimize surprise and crossfire directly. The agent tracker and in-game observation can be used to observe the reasons for the outcome, and suggest other indicators to be studied.

4.2 *3-D Game Mechanics*

The influence of *game mechanics* was in general problematic, in the failure of many experiments to isolate the influence of the controlled variables. The variety of weapons available tended to interact strongly with certain tactics and situations, overwhelming the influence of many leader decisions. When weapon choice was placed among optimization parameters, it tended to predominate as a factor in the final outcome, particularly when combinations of tactics led to consistent encounters at long or short range, and between large or small groups. The fine-grained game world also posed a variety of difficulties for agent behaviors, when tended to affect their ability to travel in close groups, the timing of their actions (as opposed to their decisions), provide backup without encountering other trouble, and in general their ability to put weapons on target.

Whereas more sophisticated behaviors could account for each of these obstacles, it would require substantial work on aspects of behavior that do not speak directly to C2 considerations. On the other hand, the approach is ideal for *human participants*, whose every movement, action, and communication would be recorded by the same instrumentation used for automated agents. Human players would be more proficient at dealing with novel circumstances, and at compensating for game idiosyncrasies, such as the particular advantages of a certain weapon. C2 configuration constraints, such as communication nets, could be enforced as for automated players. Mixed teams could also be used, with humans commanding automated agents, or following their commands. This would require some extra development effort for in-game expressibility, but would allow for the optimal mix of realistic human behavior with automation of simpler tasks. Development efforts could be optimized here by not attempting to tackle the more nuanced problems of human decision making.

4.3 Real-time Experimentation

As a consequence of using a game platform meant for human players, the time to conduct experiments was nontrivial, taking approximately four days to run only two genetic generations for the second analysis phase. Each objective function evaluation took about 1.23 hours on average.

The possibility of running the game faster than real time was not explored, but is likely to require great effort, or be impossible, in the context of a COTS 3-D game. The use of such a game lends itself to analyst-in-the-loop experimentation, where analysts examine the results of a small number of runs (perhaps by observing the game, or participating), and feed back findings and hypotheses into further small sets of runs. Large-scale Monte Carlo analysis, with hundreds of unattended runs, may be appropriate or necessary for some exploratory steps, but is unlikely to be the most fruitful technique. On the other hand, the high fidelity of any single run, along with the possibility of human participation, yields a rich domain in which the analyst can grasp tactical situations and quickly relate quantitative results (such as kill ratios) to observed qualitative phenomena (such as the effectiveness or risk of synchronized crossfire in a particular hallway).

4.4 Scenario Scope

Our testbed in this effort limited itself to the mission domain of squad-based light infantry tactics in urban firefights, with command and communications hierarchies of limited depth. While we addressed issues of command (by abstracting the effects of command configurations under casualty, within such behaviors as ‘promotion’) and communications (through the routing and reliability of radio messages), many of the more profound aspects of C2 decision making could be better addressed with further work.

Commander’s intent is one crux of C2 theory, and was addressed in this testbed only in terms of mission configuration, such as whether a team was in offensive, search, or defensive mode. To achieve more profound analysis, automated agents would require more explicit mental representations of the situation, so that analysis could be performed with respect to the accuracy and propagation of information about the battlefield and the evolving fight, toward the ideal of a common operating picture among all participants. With that capability it would be possible to examine the optimality of information sharing and the distribution of command decisions. Different extents and kinds of information flow from ground to commander and back again could be examined. Furthermore, different schemes of delegation of decision authority (possibly all the way down to the absolute ‘edge,’ the individual soldier on the ground) could be explored, with emphasis on the execution of a commander’s intent that may change over time in response to unfolding events. Mental contents and processes of each agent would contribute to the data stream, allowing for analysis of mission outcome not only on the basis of kinetic events, but also, for example, on the accuracy and precision of the commander’s (or any other player’s) mental picture of the situation.

Network-centric warfare in this testbed is limited to the simple and well established technology of basic infantry radio. The game engine allows for the implementation and analysis of many other awareness and communications devices, such as unmanned aerial vehicles (UAVs), infrared goggles that can see through walls, and personal GPS-based

position tracking systems such as the ‘blue force tracker’ (BFT) used in tanks. With full access to game physics, these devices are straightforwardly implemented, and could be carefully calibrated to replicate realistic performance, such as requiring line of sight and a minimum resolution for UAV cameras, and varying the penetration of infrared for different materials. Here, the effects of different technologies on tactics and decision making could be studied; individual soldiers could have access to a sort of RADAR drawing data from UAVs and BFTs, and could be outfitted with the goggles. Alternatively, HQ commanders could have access to such data, as from UAV operators, and could make decisions and disseminate this information to soldiers on the ground. Issues of optimality of the amount of information passed, as well as the allocation of the burden of analysis of this information into more usable products, could be studied.

5 Conclusion

The C2 testbed in its current form can be used for extensive analysis to greater extend and with different emphases than the preliminary analyses we performed.

Further development could include the implementation of an off-scene commander agent, who could be configured to communicate with various participant agents and issue orders to them. This would establish the influence of a common operating picture, or complete knowledge about the ongoing state of the game, which the commander would attempt to synthesize as a basis for decisions. This would allow for analyses of the ideal communications content, level of control by the commander, and distribution of decision-making responsibility over all agents, in order to determine optimal empowerment of the ‘edge’, or the players on the ground, while still making maximal use of available information.

The primary advantage of this testbed, the robustness of the game domain and the richness of the resulting data, is also its downfall for certain purposes. There are simply too many contingencies to allow for the development of agent behaviors in a reasonable amount of time, at the level of fidelity of the urban firefight. While we did demonstrate the feasibility of experimentation with some simple C2 ideas, a good deal of analytic effort is spent ridding the results of the idiosyncrasies of agents’ negotiation of game mechanics. We recommend the continuation of this approach in two primary ways:

More human participation, making full use of the robustness of the domain without expending the development effort of developing behaviors for every possible contingency in a high-fidelity environment. Automated agents could be mixed with humans on teams and be used for basic tasks, which would preserve the advantage of computational insight into the agent’s cognition. Further development of agent mental modeling would be fruitful in allowing the experimenter to measure the accuracy of an agent’s picture of the battlefield, possibly as based on information disseminated by human participants. This approach would also allow development to focus on the implementation of NCW devices, such as UAVs or infrared goggles, which programmatic access to the game engine makes feasible and straightforward. Although the mental pictures and decision making process of human participants would be ‘black boxes,’ having no contribution to the instrumented data stream, there may be other ways to collect their experiences out of band, such as through interview or by direct participation in the analysis of results.

A lower fidelity domain, which would allow for fully automated players with behaviors orienting on collecting, processing, and disseminating information, and making decisions, rather than negotiating a high-fidelity physical environment. A strategy game, played on a game board rather than a 3-D environment, is a good candidate. This would vastly reduce the state space of the game, and so reduce the amount of data available for the mining of interesting results, but would also allow for more complex behaviors to be implemented and more extensive analyses to be performed. Of course, the advantages of human participation remain, but analysis would be possible with fully automated players, and with sufficient control over the code base of the simulation, could be run far faster than real time. This would be ideal for the genetic algorithm approach that we applied in our preliminary analysis.

6 References

- Alberts, D. S., Garstka, J. J., and Stein, F. P. "Network-Centric Warfare: Developing and Leveraging Information Superiority." CCRP, 2nd ed. (Jul 02).
- Alberts, D. S. and Hayes, R. E. "Power to the Edge: Command and Control in the Information Age." CCRP (Jun 03).
- Fu, D., Houlette, R., Jensen, R., and Bascara, O. (2003). "A Visual, Object-Oriented Approach to Simulation Behavior Authoring," in Proceedings of the Industry/Interservice, Training, Simulation & Education Conference (I/ITSEC 2003).
- Fu, D., and Houlette, R. (2002). Putting AI in Entertainment: An AI Authoring Tool for Simulation and Games. IEEE Intelligent Systems, August 2002.
- Rushing, J., Ramachandran, R., Nair, U.J., Graves, S.J., Welch, R., Lin, H. (2005). ADaM: A Data Mining Toolkit for Scientists and Engineers. Computers and Geosciences, Volume 31, issue 5, pages 607-618 (June 2005).
- Stottler, R., S. Lackey, J. B. Kirby. (2004). "Formalized Behavior Models for MOUT OPFOR Individual Combatant Weapon Firing," Proceedings of the Industry/Interservice, Training, Simulation & Education Conference (I/ITSEC 2004).