NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

AN EXPLORATORY ANALYSIS ON THE EFFECTS OF HUMAN FACTORS ON COMBAT OUTCOMES

by

Wan Szu Ching

March 2002

Thesis Advisor: Thomas W. Lucas
Second Reader: Stephen C. Upton

Approved for public release; distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

{PRIVATE }{PRIVATE } **REPORT DOCUMENTATION PAGE**	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2002	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: An Exploratory Analysis on the Effects of Human Factors on Combat Outcomes

5. FUNDING NUMBERS

6. AUTHOR(S)
Wan Szu Ching

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

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

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

The ongoing revolution in military affairs is transforming the nature of warfare. Modern combat systems are increasingly more effective yet more complex to operate. Nonetheless, their complexities cannot be compared to human behaviors—which remain the most important factor in combat. Within Project Albert, an agent-based model called SOCRATES has been developed to enable users to explore the emergent behaviors of the agents. A deep operation scenario is developed to explore the effects of human factors on combat outcomes. Two experimental designs are used in this investigation: A Latin Hypercube and a Full-Factorial Design. Using the computing facilities at NPS, MITRE and MHPCC (Maui High Performance Computing Center), a total of 174,960 runs are made. The data suggest the existence of emergent patterns, and provide some insights into the question of how much more capable a smaller force must be in order to effectively battle a larger force. In addition, the analysis shows that the Latin Hypercube Design is able to identify the same significant factors in the scenario as are obtained by the Factorial Design, but with much fewer runs.

14. SUBJECT TERMS
SOCRATES, Agent Based Simulation, Simulation, Human Dimension of Combat, Combat Models

15. NUMBER OF PAGES 103

16. PRICE CODE

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

NSN 7540-01-280-5500 | Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited.

AN EXPLORATORY ANALYSIS ON THE EFFECTS OF HUMAN FACTORS ON COMBAT OUTCOMES

Wan Szu Ching
Major, Singapore Armed Forces, Army
B.Eng., National University of Singapore, 1992

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 2002

Author: Wan Szu Ching

Approved by: Thomas W. Lucas
Thesis Advisor

Stephen C. Upton
Second Reader

James N. Eagle
Chairman, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The ongoing revolution in military affairs is transforming the nature of warfare. Modern combat systems are increasingly more effective yet more complex to operate. Nonetheless, their complexities cannot be compared to human behaviors—which remain the most important factor in combat. Within Project Albert, an agent-based model called SOCRATES has been developed to enable users to explore the emergent behaviors of the agents. A deep operation scenario is developed to explore the effects of human factors on combat outcomes. Two experimental designs are used in this investigation: A Latin Hypercube and a Full-Factorial Design. Using the computing facilities at NPS, MITRE and MHPCC (Maui High Performance Computing Center), a total of 174,960 runs are made. The data suggest the existence of emergent patterns, and provide some insights into the question of how much more capable a smaller force must be in order to effectively battle a larger force. In addition, the analysis shows that the Latin Hypercube Design is able to identify the same significant factors in the scenario as are obtained by the Factorial Design, but with much fewer runs.

THIS PAGE INTENTIONALLY LEFT BLANK

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank Professor Thomas W. Lucas for his persistent guidance and his combat modeling insight through this entire effort, from the initial thesis proposal to the completion. I would also like to thank Professor Susan Sanchez for her advice on experimental design and analysis, as well as Professor Samuel Buttrey, who was ever ready to provide his statistical insights and S-Plus expertise for my analysis.

I would like to extend a special thanks to Dr. Alfred Brandstein at Marine Corps Combat Development Command and Dr. Gary E. Horne at MITRE. Not only did their valuable suggestions set the direction of this thesis effort, they helped facilitated the data collection process at MHPCC. My special thanks also goes to Maj (Ret) Stephen C. Upton at MITRE, who kindly agreed to be my second reader and contributed useful insights on modeling with SOCRATES.

I would also like to thank Ned Bent, Dan Croghan, and Jessica Reed, from Emergent Information Technologies, Inc., all of whom provided their expertise on SOCRATES usage, which was critical to this thesis effort. Lastly, my gratitude also goes to Brent Swartz at MHPCC and John P. Kresho at MITRE, who assisted in the data generation.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

A. INTRODUCTION

The nature and execution of war is a subject that has been studied since men began organizing for battle. In 1914, the study of warfare took a significant change with the introduction of the Lanchester models. Since then, these models have served as the fundamental mathematical models upon which many modern theories of combat attrition are based, and variants are to this day embedded in many state-of-the-art military models of combat. Unfortunately, what is normally concentrated on are the easily measurable aspects of war, such as the firepower, mobility, and lethality of weapons systems. The human element, the most important factor in combat, also the most difficult to determine, is often neglected. With the increasing importance of small-scale, autonomous warfare, examining the effects of human elements on combat outcomes is essential.

The advanced warfighting concepts of the Marine Corps envisages future combat being conducted by small, highly trained, well-armed autonomous teams working in concert, continually adapting to changing conditions and environments [Ref 1]. As the models of land warfare developed thus far do not adequately represent the Marine Corps' vision of future combat, a few agent-based models are being developed under Project Albert in the hope of providing answers to questions about the uncertainties of human elements in warfare [Ref 2]. Within the Project Albert framework, this thesis uses one of the agent-based simulations, SOCRATES, to explore the parameters associated with human elements in typical Marine Corps operations.

B. MODELING SCENARIO

Using the framework provided in the paper, this thesis examines how unit cohesion affects combat outcomes with SOCRATES. A scenario, which centers around the invasion of Kuwait by Iraq some time in the future, is developed for exploration. In the scenario, a Marine Expeditionary Task Force is assigned with the mission of conducting a block in the enemy's depth as part of a larger effort to liberate Kuwait. In its amphibious assault, one of the Infantry platoons is tasked to secure a beachhead objective by capturing two of the frontal defended sectors while conducting a local block to deny enemy reinforcement (see Figure 1), and this thesis examines how unit cohesion affects the platoon's effectiveness in battling a larger force.

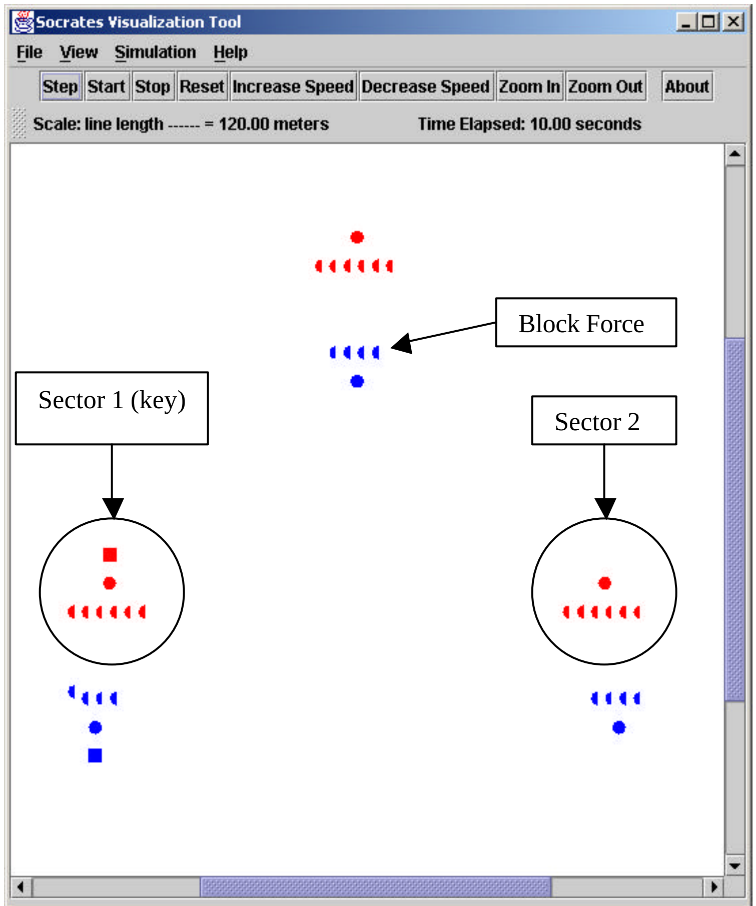

Figure 1. Start State of Scenario (Small Force Size)

This thesis models the factors attributed to unit cohesion by varying the six value components of the movement decision in SOCRATES. To study the effects of the factors, two experimental designs are used: a Full-Factorial Design and a Latin Hypercube Design. In the Factorial Design, each value component is given three levels, and together with two levels of Pk and overall force size, a total of 174,960 runs are made, with 60 replications for each combination. In the Latin Hypercube Design, two sets of 20 combinations of the parameters are generated and 30 runs were made for each set. A total of 4,800 data are collected (2 sets of 20 combinations with 30 runs, crossed by 2 sets of Pk and 2 force sizes.) Varying the level of the overall force size allows us to examine if any emergent behavior that is observed exists at different force sizes. In addition, running the scenarios using the two different experimental designs allows us to compare the utility of the designs. To ensure that outcomes obtained are intuitive, a series of preliminary runs are conducted prior to the production runs to determine the appropriate ranges of values for the Pk and the six value components to be used. Table 1 below shows the ranges used eventually, where (the components are defined in more detail in the section on Overview of SOCRATES of Chapter II):

- Cmdr Trust indicates the level of trust an agent has in his superior
- Formation dictates the desire of an agent to remain in the formation given by his superior
- tgtInWpnRng dictates the desire of an agent to keep the enemy within his weapon's range
- tgtInSnsRng dictates the desire of an agent to keep the enemy within his sensor's range
- outHostWpnRng dictates the desire of an agent to stay out of the enemy's weapon range
- cmdrInSnsRng dictates the desire of an agent to remain within the communication range of his superior

Components	Min Value	Max Value	Delta
Size (total)	38	68	30
Pk	0.1	0.2	0.1
Cmdr Trust	0.2	0.8	0.3
Formation	0.2	0.8	0.3
tgtInWpnRng	0.4	1.0	0.3
tgtInSnsRng	0.4	1.0	0.3
outHostWpnRng	0.1	0.7	0.3
cmdrInSnsRng	0.2	0.8	0.3

Table 1. Combination of Values for Farmable Inputs

C. RESULTS

The results from the two experimental designs are compared in terms of the significant effects determined. In the Latin Hypercube Design, the Akaike Information Criterion (AIC) method is the primary method used to derive a model. After the model is derived, it is simplified by selecting only terms of statistical significance. In the Full-Factorial Design, ANOVA is the main method of analysis. For the entire analysis, instead of using the raw data of 174,960 observations, the mean of every 10 replications is used in order to work within the memory capacity of S-Plus.

1. Comparison Between Latin Hypercube and Factorial Designs

The outcomes on the significant effects obtained from both the Latin Hypercube Design and the Factorial Design are summarized in Table 2 below. The multiple R-Squared value allows us to compare how much of the total MOE variation is explained by the model. The value used below is that would have been obtained if the model shown were being used to fit data from a full Factorial Design of 2 scenarios x 2 Pks x 3 levels for the six value components of the movement decision. The results show that the Latin Hypercube Design can identify the same important effects that are statistically significant and can account for a high percentage of the total sum of squares, without losing much information on the variability of the data, with far fewer runs.

Terms	MOE 1: PercentBlueKilled		MOE 2: FER	
	Latin Hypercube	Factorial	Latin Hypercube	Factorial
Size	*	*	*	*
Pk	*	*	*	*
Formation	*		*	
TgtInWpn		*		
TgtInSns	*			
OutHostWpn	*	*	*	*
Size:Pk	*			
Size:Formation	*		*	
Size:OutHostWpn		*	*	
TgtInWpn:OutHostWpn		*		
Multiple R-Squared	0.7956	0.8595	0.8312	0.8607

Table 2. Summary of Comparison of Significant Effects for the Latin Hypercube and Factorial Designs

2. Relationships Between Size, Pk, Formation And OutHostWpn Factors

As the factors of Size, Pk, Formation, and OutHostWpn are shown to have significant effects in all of the models listed in Table 2. The following are noted:

a. When the Pk is low: The majority of the Blue are killed.

b. When the formation factor equals 0.2 and when Pk is high: In the small force size scenario, the number of Blue killed is the lowest when the OutHostWpn is ***low***. The same result is obtained for mid and high levels of the Formation factor. However, when the force size is large, the lowest percentage of Blue killed occurs at the mid value of the OutHostWpn. It is also seen (result not displayed) that the lowest percentage of Blue killed occurs at either the ***mid or high*** level of OutHostWpn regardless of the levels of the Formation factor.

This observation suggests that when the overall force size is small, and if the Blue is more capable, it will be to Blue's advantage to be more aggressive in order to optimize its superior capability. However, when the total force size is large, being over aggressive may lead to a higher casualty rate for Blue than if it is not, despite its superior capability. This phenomenon might be attributed to the reason that when the overall force size is large, after each exchange of fire, there are more Red survivors who can return fire. This may indicate the existence of some emergent behaviors that only appear at a larger force size.

3. How Capable Must Blue Be?

The results provide sufficient indication that by being twice as lethal as the Red, the Blue can achieve a higher FER (relative comparison of the gaps between the number of Blue and Red killed at low and high Pk). It is also seen that at high Pk, the percentage of Blue killed, or the number of Blue killed, can be reduced by about 50 percent or more (there are a total of 16 and 28 Blue agents in the small and large force size scenarios respectively.)

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The nature and execution of war is a subject that has been studied since men began organizing for battle. As early as 1914, F. W. Lanchester introduced the Lanchester Equations (LEs) as a model of attrition in modern warfare [Ref 1]. LEs are very intuitive and therefore very easy to apply. For the simplest case of directed fire, for example, they embody the intuitive idea that one side's attrition rate is proportional to the opposing side's force level. However, LEs are applicable only under a strict set of assumptions, such as having homogeneous forces that are continually engaged in combat, firing rates that are independent of opposing force levels and are constant in time, and units that are always aware of the positions and conditions of all opposing units. Lanchester models also have some other shortcomings: they are deterministic; they require knowledge of "attrition-rate coefficients," the values of which are very difficult to obtain in practice; and they are not able to directly account for terrain and suppressive effects of weapons. Despite their shortcomings, Lanchester models have served as the fundamental mathematical models upon which many modern theories of combat attrition are based, and variants are to this day embedded in many state-of-the-art military models of combat.

Unfortunately, what is normally concentrated on are the easily measurable aspects of war, such as the firepower, mobility, and lethality of weapons systems. An aspect of war, which is more difficult to determine, is the expected effectiveness of a unit. While a unit's effectiveness depends heavily on its equipment, such as aircraft, tanks, or ships, it can also be affected by the human elements that comprise the unit. Predicting battlefield performance is exceedingly difficult because it depends so heavily on human behavior. The human element is often the most uncertain, yet important, factor within any combat system. This is particularly so in the advanced warfighting concepts of the Marine Corps, which envisage future combat to be conducted by small, highly trained, well-armed autonomous teams working in concert, continually adapting to changing conditions and environments [Ref 1]. As the models of land warfare developed thus far do not adequately represent the Marine Corps' vision of future combat, a few agent-based models are being developed under Project Albert in the hope of providing answers to questions about the uncertainties of human elements in warfare [Ref 2]. Within the Project Albert framework, this thesis uses one of the agent-based simulations, SOCRATES, to explore parameters associated with human elements in typical Marine Corps operations.

A. BACKGROUND

Project Albert is a United States Marine Corp (USMC) project initiated to investigate the intangible factors of combat that impact on a commander's decision process. The purpose of Project Albert is to identify emergent behavior by developing models using a bottom-up approach, rather than the traditional top-down approach. By employing a bottom-up approach the emergent behavior caused by the synergy of the entities may be observed. Project Albert attempts to address three areas that conventional models, such as JANUS, are incapable of handling. These areas are

a. **Non-linear Behavior:** This is where a small change in the model baseline creates a disproportionate response. Areas of non-linear behavior can be equated to opportunities and weaknesses within a military operation.

b. **Co-evolving Landscapes:** The battlefield is always changing as each commander adjusts his plan to the changing circumstance of the battle. Co-evolving landscapes attempt to rationalize the "I think; he thinks" game.

c. **Intangibles:** Through the use of personality based agent-based models, interaction akin to the intangibles of morale, discipline and training can be observed. Project Albert attempts to use personality-based models to investigate these issues.

Thus far, Project Albert has developed a series of models aimed at providing different layers of complexity, e.g., ISAAC and SOCRATES [Ref 2]. The USMC and New Zealand Army have used the Project Albert models in investigating the intangible elements of combat, such as training and morale. In addition, initial indications are that these models have the potential to contribute to the development of Australian Army doctrine and decision support processes. To further the development of the Project Albert suite of models, Project Albert has been proposed to transit from the "development phase" to the "use phase." It has been recognized that the most appropriate means to further the development of the Project Albert models is to use them, specifically SOCRATES, to investigate "real world" issues.

SOCRATES is a distillation that was developed with the primary goal of producing a fast-running representation of ground combatants to explore emergent patterns in an urban environment. With ideas adopted from BRAWLER, a model that is used to study the air-to-air combat (aircraft and missiles), as well as TRACES, which uses the BRAWLER methodology in helicopter air-to-air combat, SOCRATES uses a value-driven decision logic in an agent-based model that can capture human factors. SOCRATES is easy to use, modify, and can trace all the decisions made in the simulation. In addition, its ability to operate in a data-farming environment enables iterative processes to be conducted to explore the effect of the model's parameters. In SOCRATES, every agent of the three levels of unit hierarchies will exhibit cooperative behavior based on some value component sets, such as survival, trust, mission accomplishment, threat attrition, obedience to order, and unit spacing. These value component sets will drive the various levels of the decision-making processes. The three hierarchies in the command and control structure, from bottom to top, are

a. Frontline agent:
   i. Movement
   ii. Employment of Weapon
b. Tactic0 Leader (e.g. squad leader):
   i. Maintenance of Formations
c. Tactic1 Leader (e.g. platoon commander):
   i. Accomplishment of Missions

The data-farming inputs include characteristics of hardware (e.g., weapon's p-kill, ranges, sensor's ranges and detection duration, movement speed of agents, etc.), value component factors for decision-making (e.g., commander trust, tactics, weights, importance multipliers of the various component factors), and scenario settings (e.g., initial positions and mission description). At the end of a run, a list of data-farming outputs is produced. Being easily converted to comma-delimited (CSV) file format, the output can readily be analyzed using statistical software, such as S-Plus. SOCRATES also provides the means for users to view the playback of the scenario as well as to perform graphical analysis.

B. OBJECTIVE AND SCOPE OF THE THESIS

The thesis has two main objectives:

a. To see how SOCRATES can be used to analyze the effects of human factors on battle outcomes. This will also serve as part of the evaluation process for the newly developed SOCRATES model.

b. To examine the effects of human factors in various scenarios.

Using the framework from Russell's paper on human capital [Ref 3], this thesis explores the effects of human behavior in war. In particular, unit cohesion, which is defined in the paper as the bonding of members of a unit or organization in such a way as to sustain their will and commitment to each other, their unit, and the mission, are explored in depth. This includes running some scenarios in SOCRATES to see how the following factors affect unit cohesion:

a. Discipline

b. Communication—Vertical and Horizontal

c. Leadership

d. Initiative

e. Trust

f. Homogeneity of Unit

The primary difficulty encountered in the analysis was how to model the human elements in SOCRATES. The present stage of SOCRATES development only allows explicit modeling of three of the above factors (commander trust, communication, as well as homogeneity of unit). The other three—discipline, leadership, and initiative—have to be modeled implicitly by varying some of the decision-value components of the movement decision simultaneously. Rather than using all value components of all decision components (movement/employment of weapons, tactic0, tactic1), only the value components of the movement decision are used in this thesis, since the movement decision is the only decision that every agent must make in SOCRATES. For example, the initiative of the soldier is modeled by varying the parameters in such a way that he has to balance between the need to follow his commander's order to move in formation and the need to stay alive using local information, such as the enemy's location, he obtains through his own sensor and the battlefield updates he receives from communication broadcasts.

However, some conflicts arise when more than one factor are to be modeled simultaneously. For example, the component tgtInWpnRng must be set high for high initiative but low for low discipline. To resolve these conflicts, the following approach is adopted: for all scenarios, all the value components contributing to the movement decision are varied from 0.2 to 0.8, and the output was examined for trends that can be mapped to each of the human factors.

In all of the scenario runs, Red is made to be superior in size than Blue, but the ratio of Red to Blue is kept constant. The purpose is to enable us to determine how Blue, being a smaller force, can be an effective force against Red through better unit cohesion and enhanced combat capability. It is also hoped that the series of runs can enable us to roughly estimate how capable Blue must be to at least match Red, if not to win, and whether the answer depends on the overall force size. The personalities of the Blue agents are varied by the different settings of the priorities in the value components of the movement decision, in order to simulate different levels of unit cohesion, as explained in Chapter III. The Measures of Effectiveness used in this thesis are:

a. Percentage of Blue Killed

b. Fractional Exchange Ratio (FER)

The two MOEs reciprocate each other. They provide a good measurement of the effectiveness of a unit, an effective unit being one that kills more enemies while keeping it own losses low.

C. ORGANIZATION OF THE THESIS

Chapter I of this thesis introduces the thesis. It provides the background, the objectives, and the scopes of the thesis. Chapter II gives an overview of the modeling tool used, SOCRATES, and elaborates some features that the thesis uses in the study of human factors, in particular those specified in the scope section. Chapter III highlights a few observations from the initial trial runs using SOCRATES and discusses how the scenarios to be studied are simulated in SOCRATES. Chapter IV then analyzes the results obtained from the runs and derives some relationships between the human factors and the battle outcomes. Finally, Chapter V draws possible conclusions and makes recommendations on how the study can be carried further, as well as how SOCRATES can be improved to enable more effective future analyses.

II. MODELING TOOL

A. INTRODUCTION

SOCRATES is a distillation being developed within Project Albert with the primary goal of producing a fast-running representation of ground combatants to explore emergent patterns in an urban environment. With ideas adopted from BRAWLER [Ref 4], a model used to study air-to-air combat (aircraft and missiles), as well as TRACES, which uses BRAWLER's methodology in helicopter air-to-air combat, SOCRATES uses value-driven decision logic in an agent-based model that can capture human factors. SOCRATES is easy to use, to modify, and to trace all of the decisions made in the simulation. In addition, SOCRATES's ability to operate in a data-farming environment enables iterative processes to be conducted to explore the effects of the model's parameters. This chapter provides an overview of SOCRATES and its value-driven decision process.

B. OVERVIEW OF SOCRATES

1. Implementation

SOCRATES is written in JAVA, as its classes support an object-oriented implementation in JAVA. The selection of objects is based on similarities in how the data is used in the decision algorithms and how information is passed between the decisions and the physical models. The resulting objects are a blend of physical world analogs and algorithm abstractions. Particular attention in the class design has been paid to the ability to modify the set of decisions performed by an agent, the set of available physical systems with which the agent interacts, and in some cases components of the decisions themselves.

2. Model Environment

The battlefield in SOCRATES is represented on a two-dimensional lattice of discrete sites [Ref 5]. Each site of the lattice may be occupied by an agent of any type. SOCRATES does not explicitly model terrain. The type of terrain to be modeled, for example, whether the terrain is open or closed, is modeled implicitly by setting the speed of the agents. However, SOCRATES does provide the capability to model obstacles in the form of obstructions that impede the movement of the agents.

Currently, SOCRATES can model forces of up to seven different sides, one of which is non-combatants. Represented in different colors, each side has three hierarchies of units, with each hierarchy represented by different symbols (see Figure 5). In a SOCRATES scenario, each side is given a mission, either a Travel, Search or Vector mission, and every agent is assigned both physical and intangible attributes. All of these inputs are contained in the data input file, together with the user-specified start state.

3. Value-Driven Decision Agents

The three hierarchies of units in SOCRATES (Figure 2) are as follows:

a. **Frontline Agent:** A frontline agent is the lowest level agent in the hierarchy. He is able to make movement and weapon/target decisions (employment of weapon).

b. **Tactic0 Leader:** Equivalent to a section commander, this leader commands the frontline agents assigned to his command and is able to make decisions on the movement formation (tactical formation decision, or tactic0 decision) for his section, in addition to his own movement and employment of weapon.

c. **Tactic1 Leader:** As the overall commander, he commands the tactic0 leaders directly and can make decisions (mission formation decision) on how his sub-units should be maneuvered to achieve his mission, in addition to those decisions that are made by the frontline agents and tactic0 leaders.

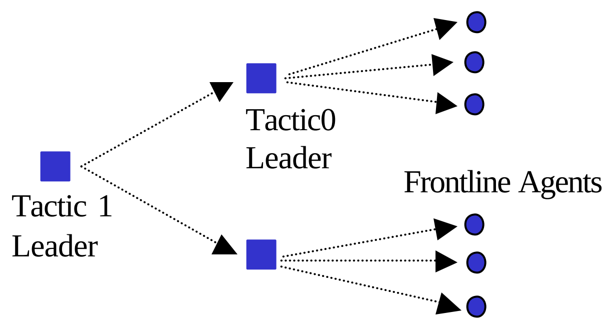

Figure 2. Unit Hierarchy in SOCRATES (From: [Ref 4])

In SOCRATES, each agent is given some attributes, both physical and intangible. Moreover, the emphasis is on the intangibles, and the physical modeling is of low resolution. Physical attributes include sensor, weapon, communication device, and speed of movement.

a. **Sensor:** The sensor used is a "cookie-cutter" sensor. The sensor will detect any enemy agent or other viewable objects that are within the range of the sensor. A sensor makes its detections periodically, and the time between each detection can be controlled, i.e., via the frame time in the farmable input. The time of the first detection attempt is randomly selected between time zero and a full frame time after time zero. This is done for each sensor, and hence, all sensors detect at different times. A sensor keeps a track it detects with the agent, and the length of time a track is kept after the most recent detection can also be controlled (timeout input). After this, the track will be removed. This allows the agents to forget contacts after a certain period of time.

b. **Weapon:** The type of weapon to be modeled in SOCRATES is determined by the range, frame time (i.e., interval between which the weapon makes its effect, or equivalently the rate of fire), and the radius of kill, provided by the user. For example, an M16 rifle will have a range of about 350 meters, a frame time of one second (sustained rate), and a radius of kill of 0.2 meter. If the distance from an agent to its perceived target's position is within its weapon's range, all other agents within the circle (with a radius equal to the radius of kill) centered at the perceived target's position may be killed. This can result in fratricide. A random draw is made and compared to the weapon's probability of kill (Pk) independently for every agent within the circle. If the Pk exceeds the number drawn, the agent is killed. This event is performed periodically at an interval equal to the frame time. In SOCRATES, any weapon can fire at any type of target, including an anti-tank weapon firing at a soldier. Therefore, only weapons of the same category should be modeled in a single scenario in order to have logical engagements. For example, small arms and weapons of larger caliber should not be used simultaneously.

c. **Communication Device:** A communication device enables an agent to pass tracks of observed enemy agents to other agents on its communication channel. Each communication device has a range and an associated list of channels, and agents on the same channel are on the same communication network. Whenever a communication device sends a message, such as an enemy track, the channel puts the message on the list of all other communication devices on the same channel, provided they are within the sender's range. To simulate communication delays, a parameter called "commInterval" is used. This interval is the amount of time that any new track must remain with the agent who observes it before he can broadcast it. In SOCRATES, no errors exist in the tracks, and all the information fuses well to form an accurate battlefield picture.

d. **Speed of Movement:** Each agent in SOCRATES is given a maximum speed at which he can move. However, in the movement decision, only the maximum speed or half of the maximum speed are considered. This feature can be used to model vehicles if so required.

In addition to the above physical attributes, every agent in SOCRATES is also given intangible attributes, such as their propensity to obey orders from higher command, aggressiveness, as well as created attributes such as initiative and leadership, which this thesis tries to model. Chapter III provides more details on how the latter attributes are being modeled.

4. Value-Driven Decision Making

The most detailed aspect modeled in SOCRATES is the agents' decision making, which employs value-driven decision logic. The flow chart in Figure 3 shows how an agent reaches a decision using the value approach. An agent receives local information via his own sensor and the information broadcasted to him via communication devices. He then updates and refines his movement model, using this and physical data, such as the sensor and weapon range, the probability of kill, etc., provided by the user. After this, a promising alternative is selected for simulation or projection. The outcome is then evaluated, and the process is repeated until the best alternative is chosen and the best decision is made. For movement decisions, which every agent makes, a total of 17 alternatives will be considered. These alternatives entail either moving at full speed along one of the eight compass directions, or moving at half speed along one of the same eight directions, or remaining stationary.

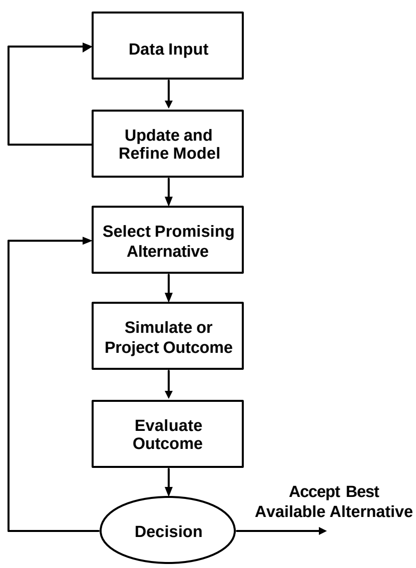

Figure 3. Flow Chart of the Value Approach (From: [Ref 4])

SOCRATES employs a multi-attribute utility function to determine the value of each alternative:

$$V(alt) = \sum_i (\text{Goal Achievement}_i * \text{utility}_i)$$

where

- V(alt) = value of alternative alt.
- i is the i^{th} value component of the movement decision.
- The goal achievement is how much the goal of the i^{th} value component is achieved.
- The utility is the maximum reward that can be obtained in achieving the goal of the i^{th} component.

The user defines the utility, which ranges from zero to one. In SOCRATES, goal achievements are measured as follows [Ref 6]:

a. **When a Threshold Requirement is Met:** The scoring function used here is a "Smoothed Step Function" (see Figure 4). A "smoothed" rather than a pure step function is used so that an agent achieving a fraction of a goal is also rewarded. This can further be divided into two categories:

i. **Threshold Reward Function:** This function rewards an input value greater than a threshold. A value component that uses the Threshold Reward Function is the "Amass" component of the tactic0 decision. A tactic0 commander is rewarded for having a locally superior force ratio. He will be penalized if he has an unfavorable force ratio. Figure 5 below shows the scoring function of this component. If the force ratio is 1:1 and the width is 0.63 (chosen so that a force ratio of 3:1 will achieve a reward of 0.8), the reward will be zero. If the force ratio is 2, the reward achieved will be about 0.65. However, if the force ratio is 1:2, the reward will be -0.65.

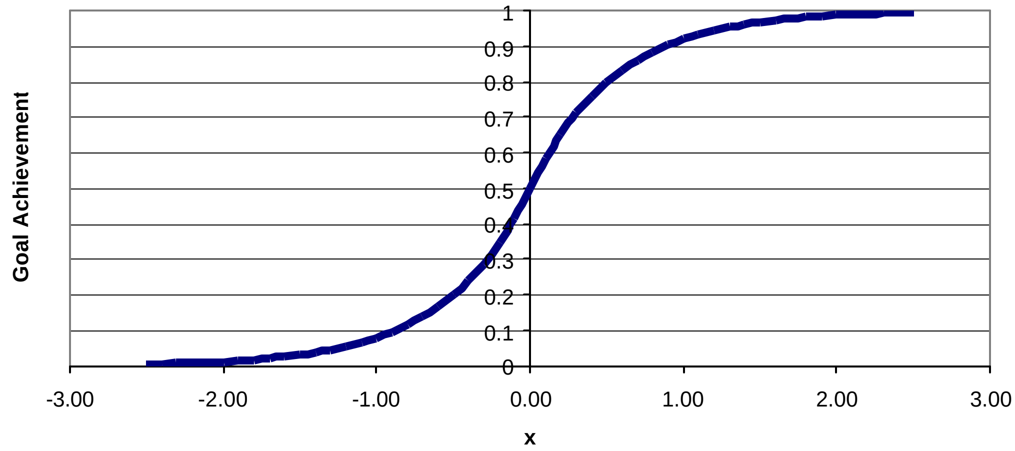

Figure 4. Typical "Smooth Step Function" (From: [Ref 6])

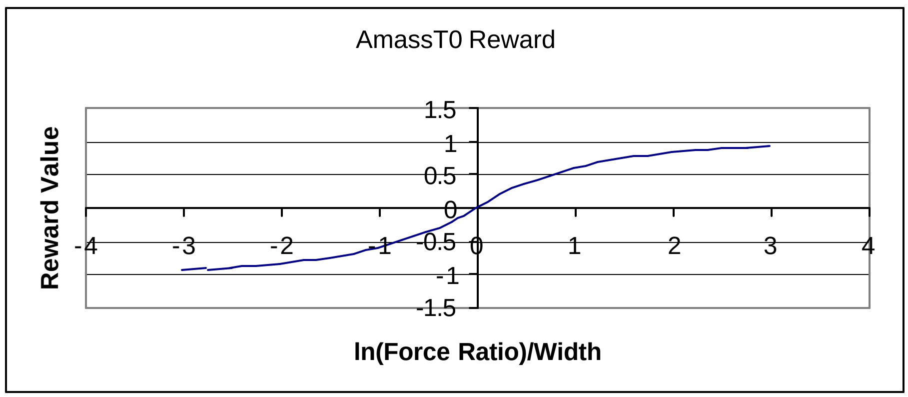

Figure 5. "Smooth Delta Function" of the "Amass" Value Component (From: [Ref 6])

ii. **Threshold Maintenance Function:** This function rewards a value that maintains its value above (or below) a threshold or rewards a value that is moving up (or down) if it is currently not above the threshold. The rewarding concept of this Threshold Maintenance Function is similar to that of the Threshold Reward Function, except that the equation used to calculate the reward is different. A value component that uses this function is the "outHostWpnRng", in which an agent is rewarded for moving out of the enemy's weapon range.

b. **When a Specific Goal is Met:** Shown in Figure 6, the "Smoothed Delta Function" is used when the goal is measured by how close an agent gets to a particular location in the goal space. The rewarding concept is similar to that of the Threshold Reward Function. The measurement of such a goal can be divided into three categories:

i. **Value Reward Function:** This function rewards a value for being nearly equal to a reference value. Only used in the engagement component of tactic1 decision, this function determines how well the engagement component of an alternative scores by having the agents move toward opposing agents.

ii. **Value Seek Function:** This function rewards a value for being nearly equal to a reference value and for its derivative to maintain this equality. Only used as part of the Vector Seek function, the Value Seek Function has inputs that indicate not only whether the test value is close to the reference value, but also whether the test value is moving in the direction of the reference value. It returns high numbers (near 1.0) if the test value is close to the reference value and is moving towards the reference value.

iii. **Vector Seek Function:** This function is similar to the value seek function except that it rewards a three-dimensional vector that matches a reference vector and its derivative. Value components that use this function include those that measure spatial goals, such as the formation component of movement and tactic0 decisions and the posgoal component of tactic1 decisions.

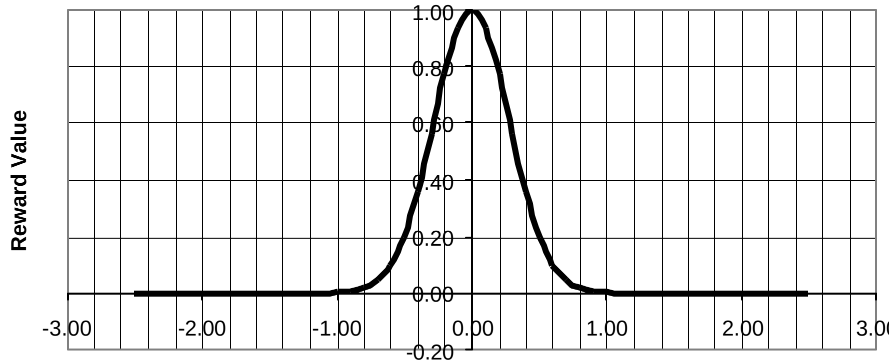

Figure 6. Typical "Smooth Delta Function" (From: [Ref 6])

Four types of decisions are made in SOCRATES, namely:

a. Movement Decision

There are a total of 17 alternatives to choose from before a movement decision is made. The alternatives include moving at full or half speed, along one of the eight compass directions, or remaining stationary. The agent has to decide at what speed and what direction he should move in order to achieve the highest total score from his movement goals (value component sets). These goals are controlled by two data elements: an importance multiplier and a width. The importance multiplier indicates the relative priority of the component, which is also the utility (see Section 4 above). The width indicates the size of the transition region where the component changes from bad to good, or vice versa.

a. **Commander Trust:** This factor is a multiplicative factor applied to other components that are related orders and communication from an agent's superior. Components affected include:

   i. The formation component of movement decision

   ii. The formation, cmdrInSnsRng, and hold components of tactic0 decisions

b. **Formation Component:** Rewards an agent for moving into the formation given by the agent's superior.

c. **TgtInWpnRng Component:** Rewards an agent for moving to within its weapon's range of the selected target.

d. **TgtInSnsRng Component:** Rewards an agent for moving to within its sensor's range of the selected target.

e. **OutOfHostRng Component:** Rewards an agent for moving out of the hostile weapon's range.

f. **CmdrInSnsRng Component:** Rewards an agent for moving to within communication range of his superior.

b. Weapon-Target Decision

A weapon-target decision involves making a decision on which target, if any, to engage. This is governed by the target value that is assigned to all agents to provide some form of priority for engagement. Given two targets that satisfy the weapon envelope equally, the one with the higher target value will be preferred. Higher-valued targets farther away may be preferred over closer lower-valued targets. This allows the user to assign high value to targets of significant importance, such as the commander, section leaders, or agents with a capable weapon, sensor, communication, or movement system.

c. Tactic0 Decision

A tactic0 decision is a low-level formation decision designed for low-level group leaders. This decision determines the location and spacing of a group of subordinates, whose purpose may be to observe and to engage the enemy, but who is not expected to be leading their own groups of subordinates. The goals, to be achieved through movement, are

a. **Formation Component:** Same as (b) for movement decision.

b. **CmdrInSnsRng Component:** Same as (f) for movement decision.

c. **Spacing Component:** Rewards a tactic0 commander for having his subordinates no closer than 1.5 times its sensor's range to the nearest friendly unit. This helps to prevent the units from overlapping.

d. **Observe Component:** Rewards a tactic0 commander for having his subordinates at the maximum sensor's range from the nearest hostile.

e. **Hold Component:** Rewards a tactic0 commander for having his subordinates moving into the hold formation given by the agent's superior. This happens only if the superior is given a TRAVEL mission.

f. **Evade Component:** Rewards a tactic0 commander for having his subordinates outside the maximum weapon's range of known hostiles. It specifies the maximum number of hostile agents that a tactic0 commander can allow to be within the weapon's range of each of his subordinates.

g. **NotBeSurrounded Component:** Rewards a tactic0 commander for having his subordinates avoid being surrounded by hostiles.

h. **Attack Component:** Rewards a tactic0 commander for having his subordinates within the weapon's range of nearest hostile. This component can be used as a factor that determines the aggressiveness of an agent.

i. **Amass Component:** Rewards a tactic0 commander for having a local force superiority. It calculates the ratio of friends to hostiles within the maximum of an agent's weapon range and rewards if the ratio is greater than one.

d. Tactic1 Decision (Mission Formation)

This level of decision is made by the tactic1 leader (highest in the unit hierarchy) so that he can fulfill his assigned mission. There are three types of missions in SOCRATES:

a. **Vector:** To move to a mission goal location without regard for the enemy.

b. **Travel:** To move to the mission goal location unless the force encounters and engages the enemy. If so, the leader must try to hold his position.

c. **Search:** To move to the mission goal location. If encountering enemies along the way, the agents will adopt a wide formation to try to see all the enemies. While approaching the enemies, the agents then tighten up their formation to gain local numerical superiority.

The type of mission selected will have an effect on the tactic1 decision:

a. **Engagement Component:** Rewards a tactic1 commander for having his troops in an engage position.

b. **PosGoal Component:** Rewards a tactic1 commander for proceeding to his goal position.

If the agents are given a Vector mission, that is to move to the mission goal position without regard for the enemy, the engagement component will always be disregarded.

5. Data-Farming

A major key feature of SOCRATES is its ability to operate in a data-farming environment. A data-farming environment enables the investigation of a wide number of variables across a wide range of values. In essence, the user is attempting to model all possible combinations and variations within the data space. The farmable inputs in SOCRATES include:

a. Hardware:

i. Ranges of sensors and weapons

ii. Sensor track timeout (the duration that a sensor keeps a track in its track bank)

iii. Pk of weapon and effective casualty radius

iv. Maximum speed of movement

b. Decisions:

i. Importance multiplier (utility), and the transition widths of all goals

ii. Tactic weights

iii. Commander trust

c. Scenario:

i. Initial positions

ii. Mission description

Data-farming relies on two key elements, reliable models and high performance computing assets.

Models suitable for use by Project Albert were not initially available. Hence, ISAAC and Einstein were developed [Ref 2], both permitting the user to define the measures of success variables within each model. The input variables are then adjusted over a specified range to generate multi-dimensional data output. More recently, ARCHIMEDES and SOCRATES were developed, with improved capabilities to model agent decision-making.

In order to exploit the full potential of agent-based models, several thousand iterations are usually run in order to produce statistically significant outputs over a range of input settings. Currently, the main resource that supports this high computing requirement comes from the Maui High Performance Computing Center (MHPCC), which has the capability to perform four billion calculations a second.

6. Outputs

Currently, for each run, SOCRATES provides a few output files, one of which is the MOE output file. This file contains 18 MOE values and the visualization output file:

a. **MOE Output File:** This is a text file that contains the results of all the MOEs that SOCRATES produced. A sample line from the MOE output file is as follows:

660.0,0,12345,"Data Farming information",10,7,-1.0,169.1582665432952, 64.93299348528853,26.123415198210786,-1.0,-1.0,574.3920172439301, 214.3920172439301,39.502117113588206,19.123415198210786,19,112

The fields are

i. Total simulation time (660.0)

ii. Index (0)

iii. Seed (12345)

iv. Data farming string ("Data Farming information")

v. Number of red dead (10)

vi. Number of blue dead (7)

vii. Time when 100% of red dead (-1.0 means it did not happen)

viii. Time when 75% of red dead (169.16)

ix. Time when 50% of red dead (64.93)

x. Time when 25% of red dead (26.12)

xi. Time when 100% of blue dead (-1.0 means it did not happen)

xii. Time when 75% of blue dead (-1.0)

xiii. Time when 50% of blue dead (574.39)

xiv. Time when 25% of blue dead (214.39)

xv. Time when first Blue agent killed (39.502)

xvi. Time when first Red agent killed (19.123)

xvii. Number of Red tactic0 decisions made (19)

xviii. Number of Blue tactic0 decisions made (112)

A SOCRATES run writes this line at the end of each run. If this file already exists, SOCRATES will append to it, thus enabling outputs of all runs of the same scenario to be saved in a single file for easy retrieving and processing.

b. **Visualization Output File:** This file contains the information necessary to run the visualization playback tool. Although not visibly clear, five different types of information are displayed:

i. **Agent Initialization Section:** This section initializes all the agents defined in the models.

ii. **Obstruction Initialization Section:** This section initializes the obstructions built in the scenario. The obstructions are composed of squares of size defined and positioned by the users, with the main purpose of obstructing movement. This section and the agent initialization section occur only at the beginning of the text file.

iii. **Maneuver Decision Section:** This section displays all the maneuver decisions that every agent makes, at the time when each decision is being made. It shows all the alternatives that the agent has, each alternative with the score of every value component, and the final score for that alternative. It also shows the final decision that the agent makes.

iv. **Agent State Section:** This section shows the states of all agents, i.e., position, orientation and whether the agent is dead or alive, at every 10 seconds of scenario time.

v. **Done Line:** Being the final line of the file, this indicates to the playback tool that the scenario file is finished.

c. **Visualization Playback Tool:** The visualization playback tool allows the user to view the movement and placement of the agents within a SOCRATES run after that run has been completed. The optional playback file (visualization output file) produced by SOCRATES records these movements every ten seconds. Figure 7 shows a typical playback of a completed run. While the blue and red icons represent the agents of the two sides, the grey are the agents that have been killed. Note: The figures in this thesis are best viewed in color.

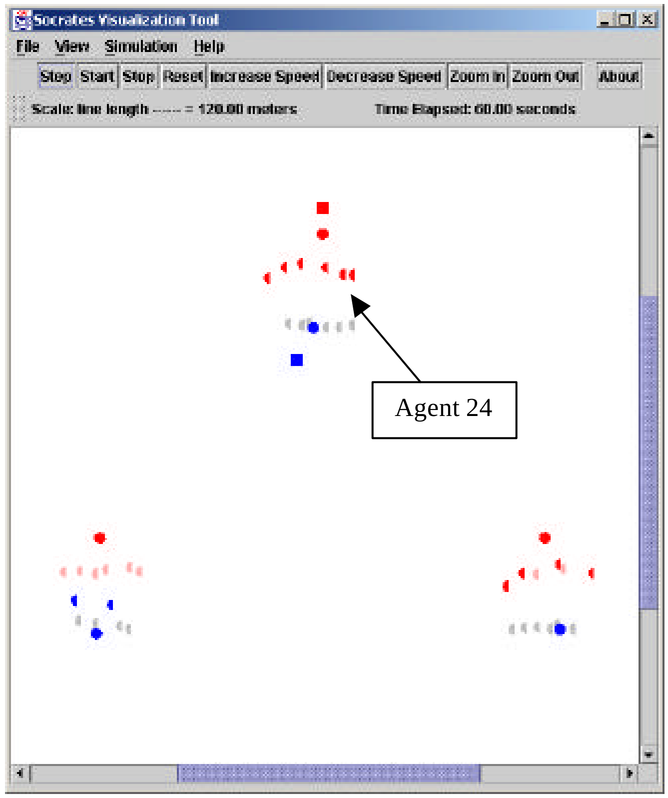

Figure 7. Visual Display on a Visualization Playback Tool

During the playback, the user can always see the movement alternatives of every agent by clicking on that agent. For example, clicking on Agent 24 in Figure 7 indicated by the arrow will reveal a screen (see Figure 8 below) that displays the 17 alternatives for Agent 24 with their respective scores. Represented by bars consisting of colored segments, each alternative suggests either to move at half or full speed along the shown direction, or to remain stationary (the bars at the center of the two groups). Each colored segment represents a particular value component (see legend at the bottom of Figure 8) of the decision, and the length of each segment is proportional to the score the value component achieved. The total score of each alternative is shown beside the alternative. Figure 8 also shows that the best alternative, with a total score of 1.61, is for the agent to move along its six o'clock direction at full speed. In that alternative, as the figure shows, the value component outHostWpnRng (out of hostile weapon range) achieved the highest score among the value components, based on the length of the grey segment. A different surface is drawn for the minimum, maximum, and average outcomes over the multiple Monte Carlo runs.

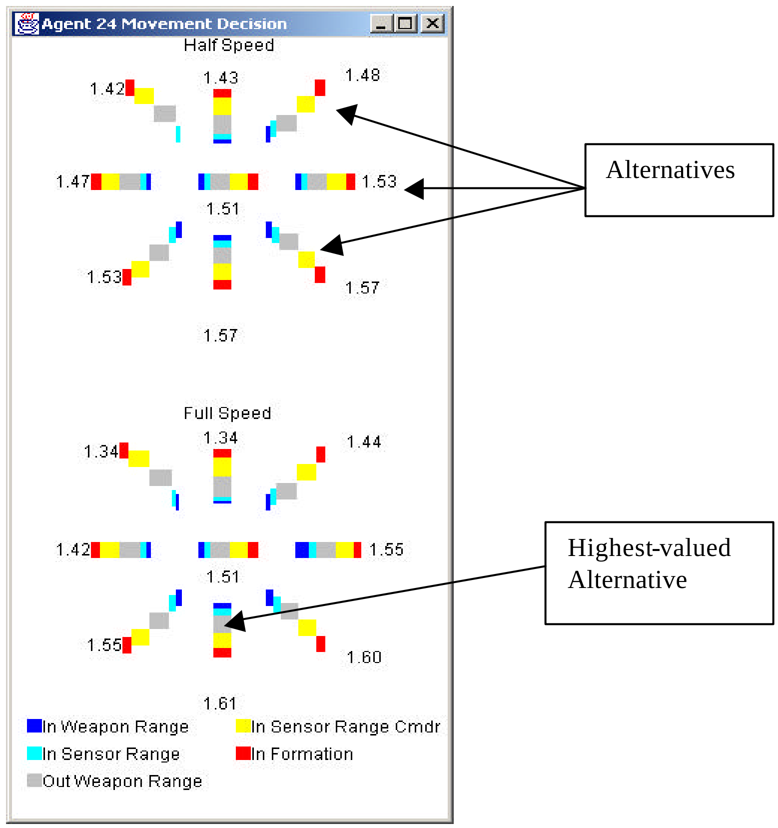

Figure 8. Showing All Alternatives of Agent During Playback

d. **Visualization Toolkit:** With this toolkit, the user can graphically, in three dimensions, view the relationship of any MOE versus any three input variables. Figure 9 shows the three-dimensional graph of the number of Blue agents killed versus the Red sensor's range and Red Pk. By adjusting the value of the third variable via a slide bar (not shown), the output will change accordingly.

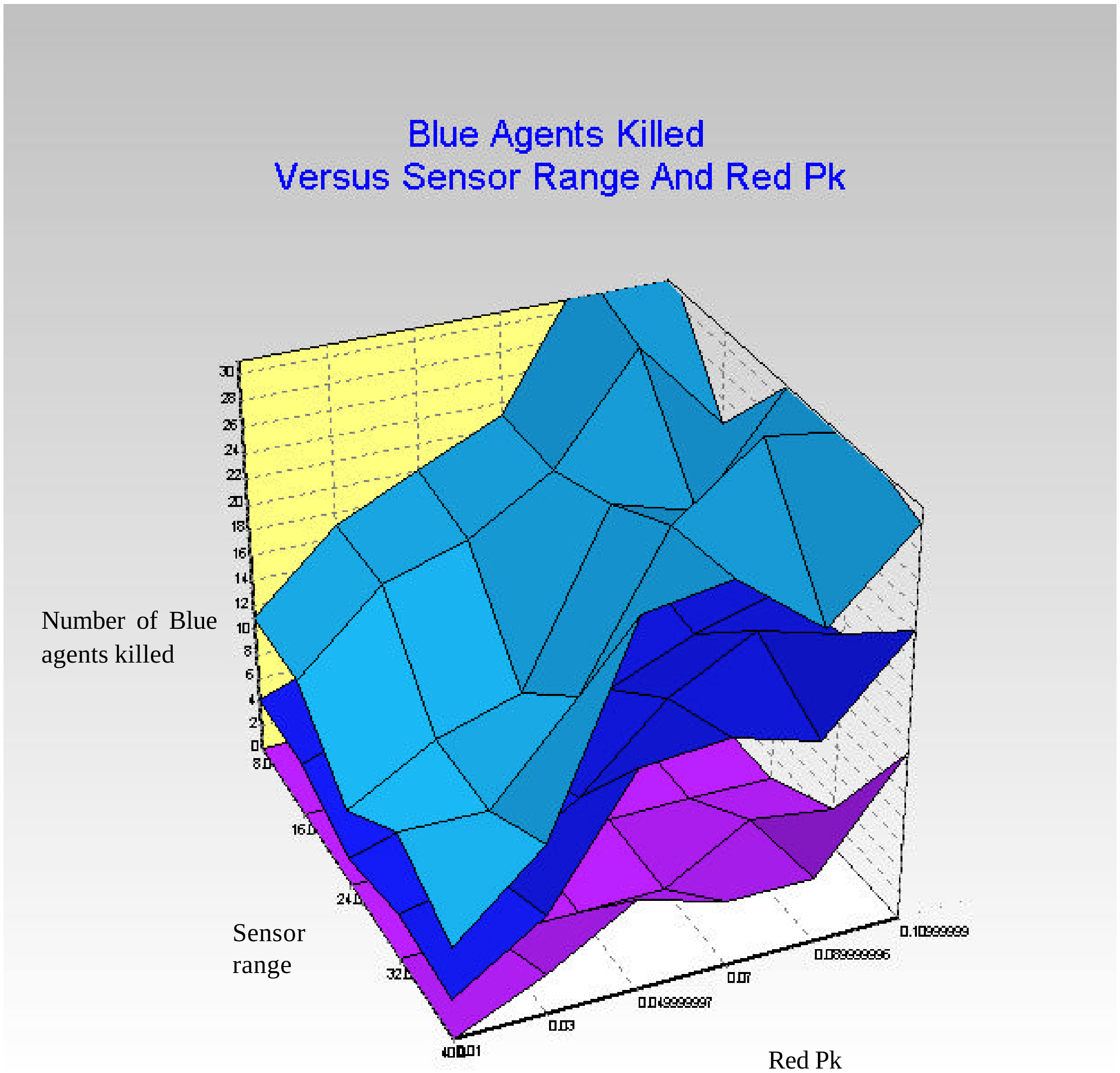

Figure 9. Typical Three-dimensional Graph of a MOE vs Input Variables

III. METHODOLOGY

This thesis aims to see how SOCRATES can be used to analyze the effects of human factors on battle outcomes and then to examine the effects of human factors in various scenarios. This chapter elaborates on the scenario setting and the methodology used to model the human factors for which SOCRATES is unable to model explicitly. The chapter also summarizes some of the problems encountered during pre-runs.

A. SCENARIO SETTING

The scenario setting used in this thesis centers around Iraq's invasion of Kuwait and Saudi Arabia at some time in the future. Faced with escalating social and political unrest, Saddem Hussein stirs up a series of activities in the name of a holy war against the western powers, hoping to strengthen his political position and to gain support from the other Arabs nations. In particular, he launches a major offensive against Kuwait and Saudi Arabia with the immediate objective of gaining control of their major oil refineries. In addition to an increase in oil prices, he hopes that a quick success will bring the Arab nations to his side, and perhaps bring about a negotiated peace that would be more to his advantage. However, the invasion is met with strong resistance from the UN forces defending Saudi Arabia, as well as the Arab Gulf Coalition forces defending Kuwait. While the UN forces successfully repel the Iraqi's attack at the border of Saudi Arabia, the Arab Gulf Coalition forces are unable to hold the Iraqi forces at the northern border of Kuwait. The Iraqi forces punch through the defense line and advance 110 km southward. Fortunately, the success of the defense by the UN forces provides sufficient time for the UN to redirect their air power to the east. Reinforced by the air power of the UN forces, the Gulf Coalition forces in Kuwait manage to halt the Iraqi's advance. In fact, the few days of continual air strikes inflict substantial casualties on the Iraqi forces greatly disrupting their resupply operations. The UN forces thus decide to exploit the situation and launch an offensive to destroy the Iraqi forces in Kuwait's territory. The counter-offensive involves three phases (see Figure 10):

a. **Phase I:** Insert a sizeable force into the enemy's depth in the north to cut off the enemy's withdrawal route.

b. **Phase II:** Conduct a two-prong attack from the south and the west to destroy the entrapped enemy.

c. **Phase III:** Mopping and flushing of remaining enemies out of Kuwait's territory.

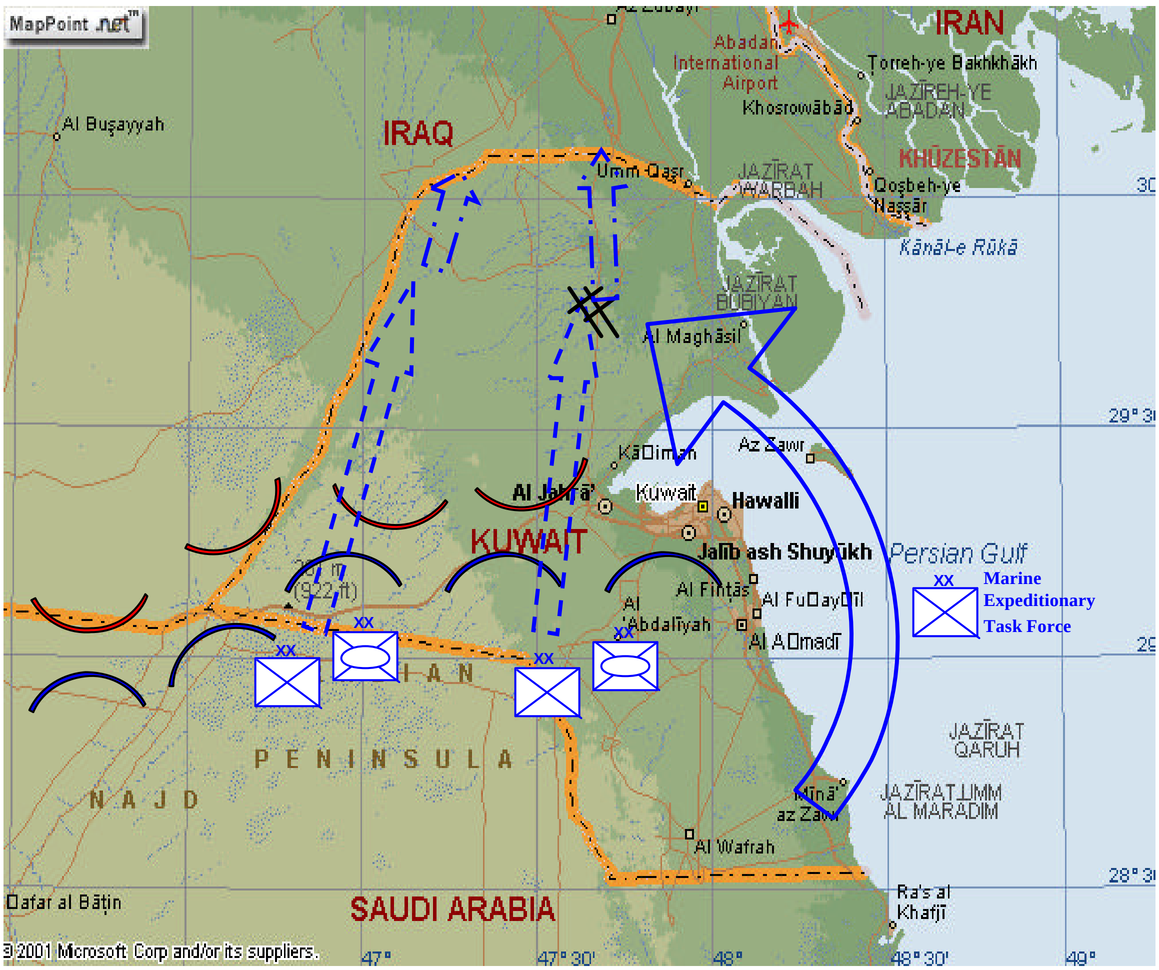

Figure 10. Scenario Setting

As part of the offensive, a Marine Expeditionary Task Force is tasked to conduct an amphibious assault at the northern beach of Kuwait to cut off the enemy's withdrawal route. In order to enable the landing of follow-on forces, a Marine Brigade is inserted by air into the enemy's depth to capture a beachhead. This thesis works on the scenario in which an infantry platoon tries to capture part of the beachhead objective by simultaneously attacking two of the enemy's defended sectors while establishing a block to deny the enemy reinforcements from the third sector.

B. METHODOLOGY

1. Modeling the Scenario

The main aim of the analysis is to examine the effects of human factors on combat outcomes. The emphasis is to consider how the agents behave in the face of the enemy's presence and fires, rather than the movement phase. Hence, when modeling the scenarios (see Appendix I for scenario file of small force size scenario), a start state in which the Red and the Blue are positioned close to each other, specifically, within each others sensor range but out of each other's weapon ranges, is given. Once the simulation begins, the agents must immediately update their peers, leaders, and commanders on the battlefield situation, and all of the agents must start making decisions on their maneuver scheme in order to achieve the overall mission.

In all of the scenarios modeled, Red is given a VECTOR mission, and Blue a TRAVEL mission. At the start state, the Red force has two of its sub-units defending two frontal sectors, and a third sub-unit as the reinforcement force in the rear (see Figure 11). The Red tactic1 leader, who is the mission commander, is located in sector 1, as sector 1 is the key objective. The Blue force has two of its sub-units trying to capture the two frontal sectors with the third sub-unit conducting a block.

To model the Red force in defense, the two tactic0 leaders in the defense sectors are made to remain stationary so that the frontline agents under their command will likely not stray too far from the designated defense location, that is, from their immediate superior. However, the third tactic0 leader, who is the reinforcement leader, can move as freely as all of the frontline agents so that he can reinforce a sector requiring assistance.

The thesis looks at two scenarios:

a. Small-sized forces, with a total of 38 agents.

b. Large-size forces, with a total of 68 agents.

In each of the scenarios, the ratio of the Red force to the Blue force is always 1.5:1. Both the Red and the Blue forces are given the same values for all parameters, except for the Blue Pk and the Blue movement decision components. While the values of the movement decision components of the Red force are fixed at values to reflect a unit with mediocre personalities, the values of the movement decision components of the Blue force are varied within the ranges shown in Table 4. As will be explained in the next section, the variation in Table 4 allows SOCRATES to model agents of different personalities. In addition to the movement decision components, the value of Blue Pk is also varied between 0.1 and 0.2. Through these variations, the thesis can examine how unit cohesion (by varying the values of the movement decision components) and capability (by varying the values of Pk) affects the overall capability of a force that is inferior in size than its adversary. The increase in the overall force size from a total of 38 agents (small force size scenario) to a total of 68 agents (large force size scenario) also enables us to determine if the overall force size affects the outcomes.

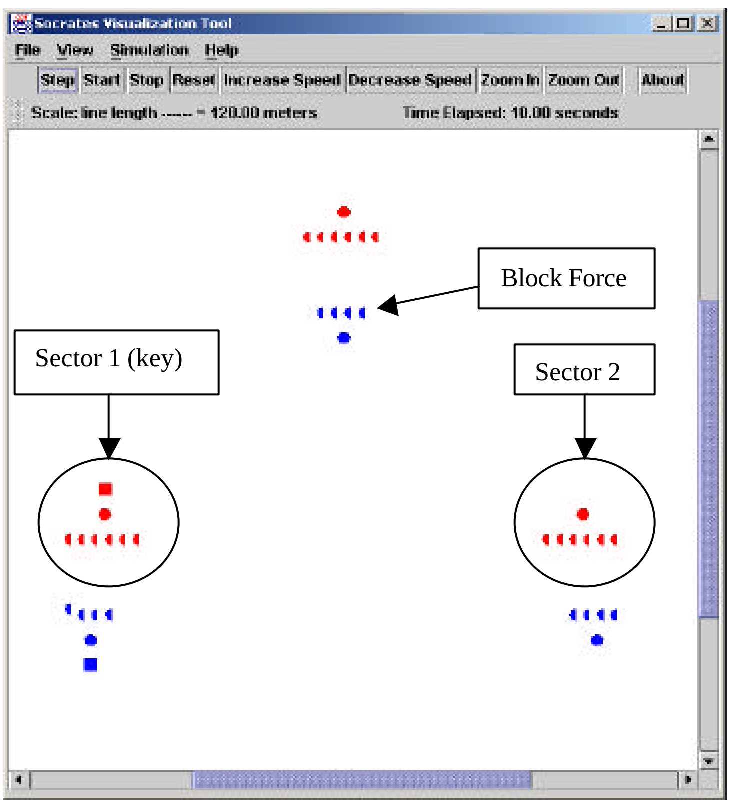

Figure 11. Start State of Scenario (Small Force Size)

2. Modeling of Human Factors

This thesis examines six factors attributing to unit cohesion:

a. Discipline

b. Communication – Vertical and Horizontal

c. Leadership

d. Initiative

e. Trust

f. Homogeneity of Unit

Of these six factors, SOCRATES can only model three explicitly: communication (via communication devices that each agent is given), trust (via the commander trust component), and homogeneity of unit (by setting the forces equally). The other three factors are composed by varying the combinations of the value components in the movement decision, as shown in Table 3 below:

Components	Cmdr Trust	Fmn	tgtInWpnRng	tgtInSnsRng	outHostWpnRng	cmdrInSnsRng
Discipline						
Low		**0.2**	**0.2**	**0.2**	**0.8**	**0.2**
Moderate		0.5	0.5	0.5	0.5	0.5
High		0.8	0.8	0.8	0.2	0.8
Initiative						
Low	0.8	0.8	0.2	0.2	0.2	
Moderate	0.5	0.5	0.5	0.5	0.5	
High	**0.2**	**0.2**	**0.8**	**0.8**	**0.8**	

Table 3. Combination of Value Components to Model Human Factors Not Explicitly Represented in SOCRATES

For example, an agent can be modeled as one of low discipline by setting the values of the parameters shown in bold under the discipline factor. Regardless of any amount of trust that the agent has in his commander, the agent who is of low discipline is envisaged to often disregard what he is being told, such as not maintaining the required distance from his commander or obeying the movement formation given (low on both formation and cmdrInSnsRng). He will also generally try to avoid any danger (low on tgtInWpnRng and tgtInSnsRng, high on outHostWpnRng), despite orders for him to close in and kill the enemy. Similarly, an agent can also be modeled as one of high initiative by the value settings shown in bold under the initiative factor. Such an agent is expected to, on his own initiative, do what he deems fit on the ground rather than blindly obeying every order his commander gives. When he sees some targets, he will make his own risk and target priority assessments. He will also decide on his own whether or not to kill the targets and will decide which target to kill first, even though he may have been ordered to move somewhere else, perhaps toward the mission goal position.

However, the above approach contains some conflicts. A conflict of interest will exist when one tries to model an agent of low discipline and high initiative, for example, which values should the components tgtInWpnRng and tgtInSnsRng take—0.2 or 0.8? Therefore, this thesis assumes a reverse-engineering approach. Rather than correlate the set of values of the decision components to the behaviors, simulation runs will be conducted by simultaneously varying the values of the components ranging from the lowest possible value to the highest. The outputs can then be mapped back to the values of the decision components to derive any possible logical patterns that can show what kinds of personalities the agents have. Such patterns, if they exist, can be used in future simulations to model agents with the appropriate type of characters.

3. Ranges of Parameters Used

While Table 6 proposes modeling three of the human factors by varying the value components of the movement decision, possibly between the range from 0.1 to 1.0, selecting extreme values for some components may cause undesired outcomes. For example, if the component outHostWpnRng (out of the enemy weapon's range) is given a high-end value, the agents will always try to avoid any enemy, and the simulation will not have any engagements. Even if there might be some engagements due to the given start state (both sides with some forces within each other's weapon range), the engagements will be minimal as the agents will break away as soon as they can Additionally, if the component tgtInWpnRng (a target in weapon range) is given a low-end value, the agents will avoid all enemies at all times. A series of preliminary runs were conducted prior to the production runs to determine suitable ranges of values that can be used. Table 4 below shows the recommended ranges of values for the various movement decision components to be used:

Components	Minimum Value	Maximum Value	Notation Used
Commander Trust	0.1	1.0	Trust
Formation	0.1	1.0	Formation
tgtInWpnRng	0.4	1.0	TgtInWpn
tgtInSnsRng	0.4	1.0	TgtInSns
outHostWpnRng	0.1	0.7	OutHostWpn
cmdrInSnsRng	0.2-0.3	1.0	CmdrInSns

Table 4. Recommended Ranges of Values of the Movement Decision Components for Intuitive Output in the Scenario

4. Measures of Effectiveness (MOE)

This thesis uses two MOEs:

a. The percentage of Blue killed (PercentBlueKilled)

b. The Fractional Exchange Ratio (FER)

The percentage of Blue killed, rather than the actual numbers, is used primarily so that the outcomes from the different scenarios, which involve varying force sizes, can be compared on a common yardstick. While the percentage of Blue killed indicates the amount of casualties, the second MOE, which is the FER, provides a sense of the effectiveness of the Blue force, given different capabilities and different levels of unit cohesion. These two MOEs (which will subsequently be addressed as the responses in this thesis although they are computed from the direct output of SOCRATES) are chosen because achieving one without the other in real operations is not desirable. A unit achieving a high FER but having a high percentage of its own troops killed will not be able to proceed to further missions.

5. Experimental Design

A Factorial Design was selected as the primary experimental method since it works well when experiments are performed to measure the effects of one or more variables on a response [Ref 7], the response in this thesis being the percentage of Blue killed. Although a large number of runs is required for the Factorial Design, the availability of the super computing power at the Maui High Performance Computing Center minimizes the time needed. For this thesis, a 2 x 2 x 3^6 factorial (or gridded) design is appropriate, as there are six value components in the movement decision of which the effect on the combat outcome is to be examined. The set of parameters to be used are as follows:

Components	Min Value	Max Value	Delta
Size (total)	38	68	30
Pk	0.1	0.2	0.1
Cmdr Trust	0.2	0.8	0.3
Formation	0.2	0.8	0.3
tgtInWpnRng	0.4	1.0	0.3
tgtInSnsRng	0.4	1.0	0.3
outHostWpnRng	0.1	0.7	0.3
cmdrInSnsRng	0.2	0.8	0.3

Table 5. Combination of Values for Farmable Inputs

A total of 60 replications were made for each set of the above parameters. This amounts to a total of 174,960 runs, as shown below.

$$\text{Total number of runs} = (3^6) * 2 * 60 * 2$$
$$= 174{,}960$$

Due to some data reading problems at Maui after their upgrading of SOCRATES to the latest version (2.2.1), a run of the small force-size scenario was made as a back up using the Virginia Cluster at MITRE, but with only 30 replications. In addition, a series of runs were also made using a Latin Hypercube design [Ref 8] with the facility at NPS.

In the Latin Hypercube run, two sets of 20 combinations of the parameters were generated and 30 runs were made for each set. A total of 4,800 data were collected (2 sets x 20 combinations x 30 runs x 2 sets of Pk x 2 scenarios). These data are analyzed separately and compared with the outputs from Virginia and Maui. We want to see if the different methods reach similar conclusions.

6. Analytical Methods

The primary tool used in this thesis is S-Plus. For the Latin Hypercube design, the result is analyzed using regression and the Akaike Information Criterion (AIC) [Ref 9] method. With the small total number of observations, S-PLUS can examine interacting effects of up to eight levels of interactions before fitting a model with the lowest residual standard error. The Analysis of Variance (ANOVA) [Ref 10] is used to analyze the effects of each of the components under examination for the Factorial Design. However, due to the memory limitation in S-Plus to handle higher order of interactions for large data sets, the mean of every 10 replications of the data for Factorial Design is used as the data input.

a. Normality Assumption of the Errors

For both MOEs, the following transformation is applied to responses in order to meet the underlying assumptions of regression [Ref 11]: Power transformation of the PercentBlueKilled of order 2 in the Latin Hypercube Design, and a logarithmic transformation of the FER by Log(FER+0.1) in both designs. The latter transformation is made so that it will be more robust against the situation in which FER equals 0, for example, when none of the Red is killed.

b. Examination of Interacting Effects

The AIC and the ANOVA techniques are then used to perform the analysis to determine the main and interacting effects for the Latin Hypercube and the Factorial Design respectively. It was noted that the AIC technique selects terms only up to two-way interactions. Thus, only two-way interactions were being examined with ANOVA in the Factorial Design. Moreover, third order interactions and above generally are difficult to interpret [Ref 12]. In order to have a data size that is manageable by S-Plus, the mean of the every 10 replications for each treatment was used for the analysis.

C. SOME INSIGHTS

In the process of conducting preliminary runs to fine-tune the scenarios and the ranges of the values of the parameters, the following insights were obtained:

1. Randomization Problem

This problem surfaced when a scenario in which both the Red and the Blue forces were equal in size and capabilities was run. It was observed that in the twelve runs made, the Red force won all the time, despite the fact that both Red and Blue were equally matched. The odds of one side winning all of the 12 battles, if the forces are truly even, is

$$2 \text{ x } (1/2)^{12} = 0.0004883$$

Thus, we concluded that something was wrong with the implementation. In the scenario used, the random number seeds chosen were consecutive, and the value of Pk was 0.8. It was later realized that the unusual phenomenon was because consecutive seeds were used.

SOCRATES uses JAVA's pseudo-random number generator. Hence, it generates random numbers that are based on the previous random numbers generated, thus generating a stream. A test program written by the program designers found that if consecutive seeds were used, it generates similar values for the second number in the stream (the seed being the first), and SOCRATES uses this second number to schedule a weapon event time for the first agent (AGENT 1). It is only after the second random number that the stream diverges. Therefore, if the Red agent is the first agent entered in the scenario input, as in the above scenario, and if the second random number so generated favors Red in such a way that it will be the first to open fire, with a high value of Pk such as 0.8, Blue will always be unlikely to survive the first round of fire. For example, if the second number generated by a seed 1234567 is 0.3, and 0.3 favors agent 1, which happens to be a Red agent, the Red agent 1 will fire the first shot. If a high Red Pk is used, say 0.8 as used in the above scenario, the Blue agents will likely be killed. As such, Red will always have the advantage, given a favorable starting condition. If the seed for the next run is 1234568, the second number generated will still be 0.3, and Red will have the opportunity to fire the first shot again. This explained why Blue always loses in all the cases mentioned above.

Further testing showed that if a random number seed of value zero is used, SOCRATES will use the system clock as the random seed. This will produce better randomization for sequential runs, since the time window for the same stream to be produced using the system clock is probably milliseconds. However, this implies that any result produced using a random number seed generated from the system clock will not be reproduceable.

2. Probability of Kill (Pk)

Because of the observed sensitivity to Pk, a separate preliminary scenario was created to see whether the magnitude of the value of Pk used would affect the outcome. In a test scenario, three Blue agents were made to engage eleven Red agents in a single file, as shown in Figure 12. The obstruction was constructed to force both the Blue and the Red agents to move toward each other instead of fanning out into formation, in which the tactic0 leader and the frontline agents would move to points where the distance between them and their tactic1 commander is 0.8 of the commander's sensor's range. A total of 80 replications were made for each sub-scenario and the results are tabulated in Table 6.

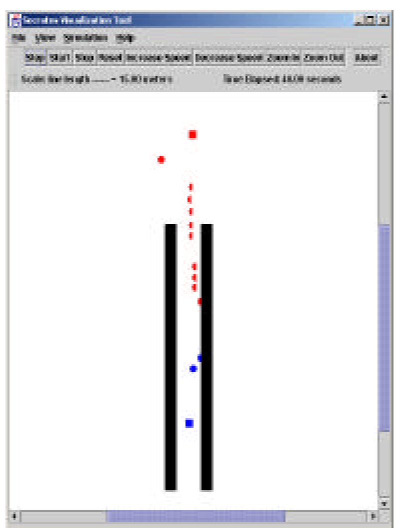

Figure 12. Scenario Used to Examine the Effect of the Pk on Combat Outcome

Sub-scenario	Red Pk	Blue Pk	Total # Red killed	Total # Blue killed
I	0.02	0.02	29	240
II	0.02	0.01	22	240
III	0.2	0.2	146	240
IV	0.2	0.1	73	240

Table 6. Outcome of Single-file Engagement Scenario

Naturally, if the Pk used is small, and when Blue and Red have the same Pk, one would expect that the Blue would have as many opportunities as the Red to return fire and to inflict more casualties on the Red than when Blue's Pk is less than Red's. This is exactly what happened in sub-scenarios I and II. Even though the Red was clearly twice as lethal as the Blue in sub-scenario II than in sub-scenario I, the total number of Red killed over the 80 runs in sub-scenario I and II did not differ significantly. However, in sub-scenarios III and IV, when Pk's of a larger order of magnitude were used, there was a clear difference in the total number of Red killed. When Red was twice as lethal as Blue, it killed Blue much faster than it was being killed by Blue and hence suffered less casualties. These results suggest that Pks above 0.1 produce more intuitive outcomes.

A related study conducted by MAJ Ashley Fry, CATDC, Australian Army, also showed that a good range for Pk is between 0.1 and 0.4. His study had two scenarios. The first scenario was a symmetric one that had the same number of Blue and Red, and both had the same capabilities and dispositions. The second was asymmetric in terms of numbers, capabilities and disposition (Figure 13 below). MAJ Fry ran the scenarios with four sets of values for Pk, as shown in Table 7 below, each with 100 replications. His output, also in Table 7, shows that the number of Red and Blue wins were approximately the same for the symmetric scenario, since the Red and the Blue were equally capable. However, in the asymmetric scenario when Red was more capable than Blue, the number of Red wins only exceeded the number of Blue wins with a comfortable margin when $Pk \geq 0.1$, and the results were consistent when $0.1 < Pk < 0.316$. This indicates the suitable range of Pk to be used. As a result of the above two tests, the Pk used in this thesis was selected to be between 0.1 and 0.4.

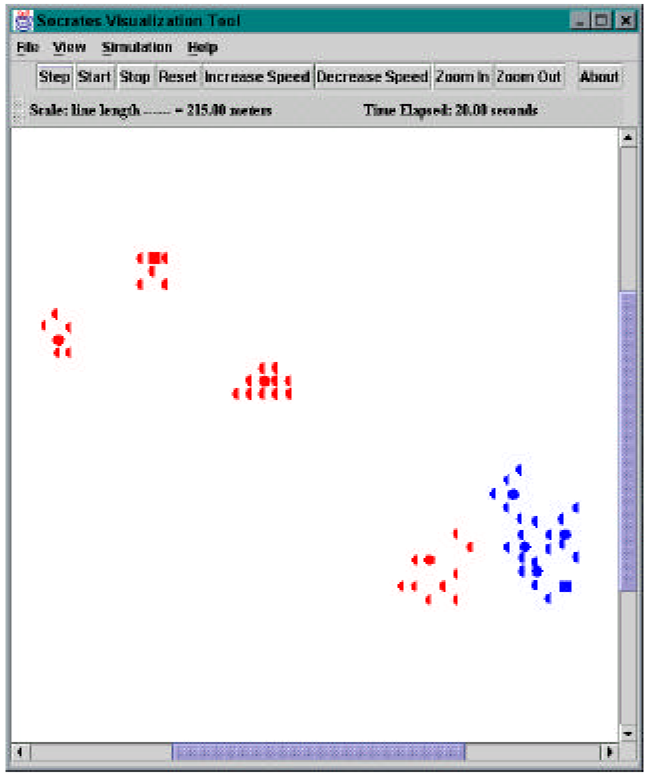

Figure 13. Asymmetric Scenario of MAJ Fry's Study

	Pk	0.01	0.0316	0.1	0.316	1
Symmetric Scenario	# Blue Wins	47	50	49	51	47
	# Red Wins	52	48	50	43	38
Asymmetric Scenario	# Blue Wins	89	30	***11***	***15***	20
	# Red Wins	8	55	***86***	***78***	71

Table 7. Outcome of MAJ Fry's Study on the Pk To Be Used

(Note: The unaccounted numbers are the ones with tied outcomes.)

3. Other Findings

The following are some other problems discovered during the preliminary runs. Their discovery helps to improve SOCRATES for future use:

a. The weapon of a killed agent continues to fire.

b. The obstruction inhibits only the movements of the agents, but not fires.

c. Whenever there is a tie between some of the movement alternatives, the first one considered will always be selected for the agent, hence a biased decision.

d. There was a data-reading problem in Maui's configuration when it upgraded its version to 2.2.1, which resulted in repetitive outputs, as the inputs were unable to be read in. The problem was discovered and rectified.

IV. RESULTS

This chapter presents the results obtained from both the Latin Hypercube and Factorial Designs. The significant effects are determined, and the results from both the Latin Hypercube and Factorial Designs are compared. It should be noted that when analyzing the data from the Latin Hypercube Design, the input variables of Size and Pk were converted to factors in S-Plus while the rest remained in numeric form. Whereas in the Factorial Design, all of input variables were converted to factors, except for the responses. In both designs, the baseline values for Size and Pk are small force size and low Pk respectively.

A. MOE 1: PERCENTAGE OF BLUE KILLED

1. Latin Hypercube Design

Figure 14 below shows the qq-plot of the model fitted on the data obtained using the power transformation of order two on the response PercentBlueKilled. The plot shows some wriggles that resemble a step function, which indicates that some of the residuals clump about the same values rather than behaving linearly, as they would for a normal distribution. The residual plot in Figure 15 reveals further that the residuals generally exhibit a decreasing trend with respect to the fitted values, that is, the errors do not have a common variance. Figure 16 depicts the corresponding histogram of the residuals. From the figure, the residuals are clearly not normally distributed, due to the existence of the two distinct spikes. The data were broken down to determine the reason for the anomalies. By breaking up the data according to the Pks, it is shown the tallest spike occurs when the residual approximately equals zero, at low Pk. In all of the scenarios used, we have a small Blue force engaging a large Red force, and hence, it is common to find a situation in which all of the Blue agents are completely killed. This is especially so when Pk is low. Therefore, when fitting a model, the "best-fit" line tends to be near large congregations of points, if not passing through them. As a result, the residuals at these points are small. This causes the tall spike at low residuals. When Pk is high, due to the superior force ratio of the Red to the Blue, there are still many occasions in which all of the Blues are totally killed, and hence, the smaller peak in the histogram of the residual around 0.3. Although the residuals are not normally distributed, the analysis will still be able to provide good indications as to which factors have significant effects on the outcomes. However, the p-values associated with hypothesis tests that assume normality will be slightly deviated from their true values.

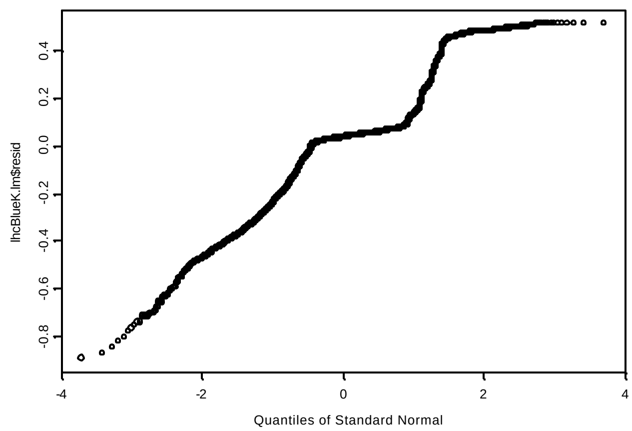

Figure 14. QQ-plot of the PercentBlueKilled Model from the Latin Hypercube Design

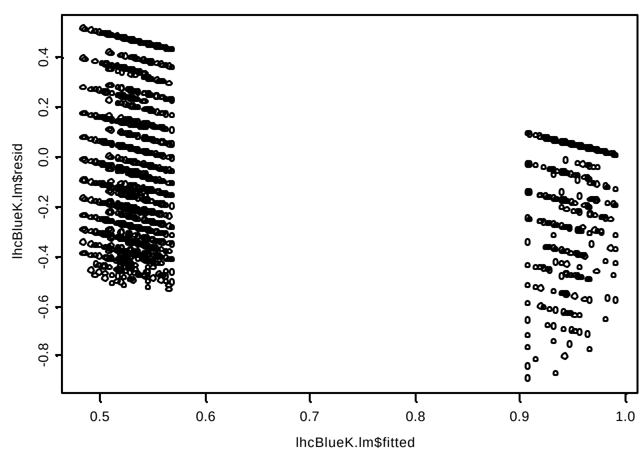

Figure 15. Residual Plot of the PercentBlueKilled Model from the Latin Hypercube Design

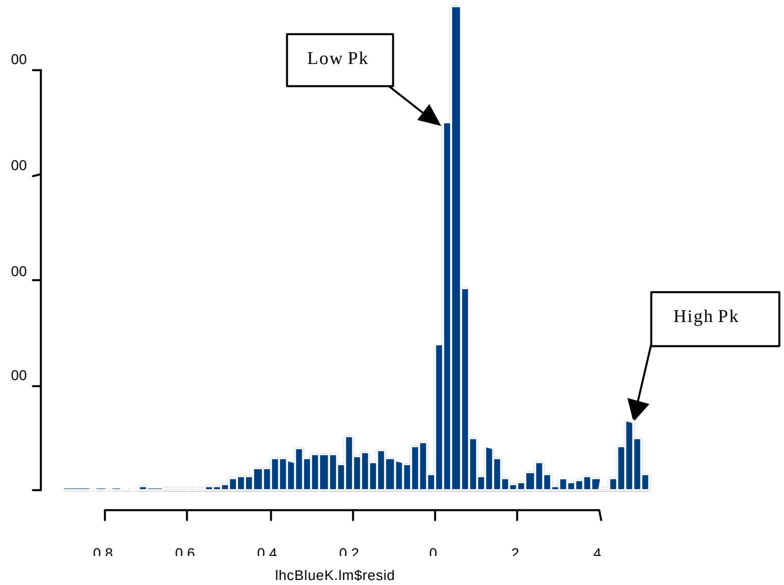

Figure 16. Histogram of the Residuals of the PercentBlueKilled Model from the Latin Hypercube Design

The model was run using the stepAIC function in S-Plus to include all interaction terms that make the AIC values low in the model. Table 8 below shows the model derived. Of the eight main effects, only five are significant at the 0.05 level—Size, Pk, Formation, TgtInSns, and OutHostWpn. In addition, all the interacting terms are also statistically significant. It is noted that the AIC method only selected terms up to two-way interaction. If we include only significant main effects and interacting terms which comprises the significant main effects in them, as shown in bold, from Table 8, we will obtain a model that has a multiple R-squared value of 0.4676, which is just a minute reduction from 0.4738. The R-Squared value allows us to compare how much of the total MOE variation is explained by the model. Hence, we can have a simpler model without losing much information on the variability.

Next, among the bolded terms, we can see that only the factor Size has a positive coefficient among the significant main effects. Since MOE 1 measures the percentage of Blue killed, one will expect to see a negative correlation between the PercentBlueKilled and the main effects, since higher values of Pk, Formation, OutHostWpn, etc., will mean more capable agents who will follow orders to move in formation, and not to stay close to the enemy. However, when the overall force size increases, Blue's casualty rate increases too, despite the fact that the force ratio between the Blue and the Red remains the same. This may be a sign for the existence of some emergent behaviors.

Coefficients:

	Value	Std. Error	t value	Pr(>|t|)
(Intercept)	1.0002	0.0515	19.4338	0.0000
Size	**0.1202**	**0.0328**	**3.6690**	**0.0002**
Pk	**-0.3881**	**0.0093**	**-41.9505**	**0.0000**
Trust	0.0785	0.0435	1.8052	0.0711
Formation	**-0.2431**	**0.0611**	**-3.9811**	**0.0001**
TgtInWpn	0.0340	0.0533	0.6383	0.5233
TgtInSns	**-0.0982**	**0.0478**	**-2.0550**	**0.0399**
OutHostWpn	**-0.1608**	**0.0491**	**-3.2780**	**0.0011**
CmdrInSns	-0.1203	0.0626	-1.9206	0.0548
Size:Pk	**-0.0709**	**0.0131**	**-5.4216**	**0.0000**
Formation:TgtInWpn	0.1861	0.0691	2.6915	0.0071
Size:Trust	-0.0632	0.0240	-2.6333	0.0085
Formation:CmdrInSns	0.1133	0.0523	2.1683	0.0302
Size:Formation	**0.0524**	**0.0240**	**2.1855**	**0.0289**
Size:TgtInWpn	-0.0766	0.0361	-2.1207	0.0340
Trust:CmdrInSns	-0.2390	0.0691	-3.4592	0.0005
TgtInSns:CmdrInSns	0.2947	0.1004	2.9365	0.0033
Trust:OutHostWpn	0.1896	0.0752	2.5200	0.0118

Residual standard error: 0.2266 on 4782 degrees of freedom
Multiple R-Squared: 0.4738
F-statistic: 253.3 on 17 and 4782 degrees of freedom, the p-value is 0

Table 8. AIC Model for the PercentBlueKilled from the Latin Hypercube Design

2. Factorial Design

The data is first examined graphically using the plot.design and plot.factor functions in S-Plus. Figure 17 shows the plot of the mean of each treatment level. This plot provides a clear initial indication that the Size, Pk, TgtInWpn and OutHostWpn have more distinguishable effects than the other factors. The effects of the Trust, Formation, TgtInSns, and CmdrInSns are minimal since the largest variation of these four is not greater than 0.5. This small variation in the percentage of Blue killed will generally not have any physical significance, especially in the context of the thesis' scope which looks at small-scale combat. The followings are also noted from the plot:

a. When the total force size is large, the percentage of Blue killed is also higher, which seems contradictory to our initial instinct which believes that the mean percentage of Blue killed should be about constant, if not better, if the number of troops of Blue increases although maintaining the same force ratio.

b. The percentage of Blue killed is high when the value of the TgtInWpn is high.

c. The percentage of Blue killed is high when the value of the Pk or OutHostWpn is low.

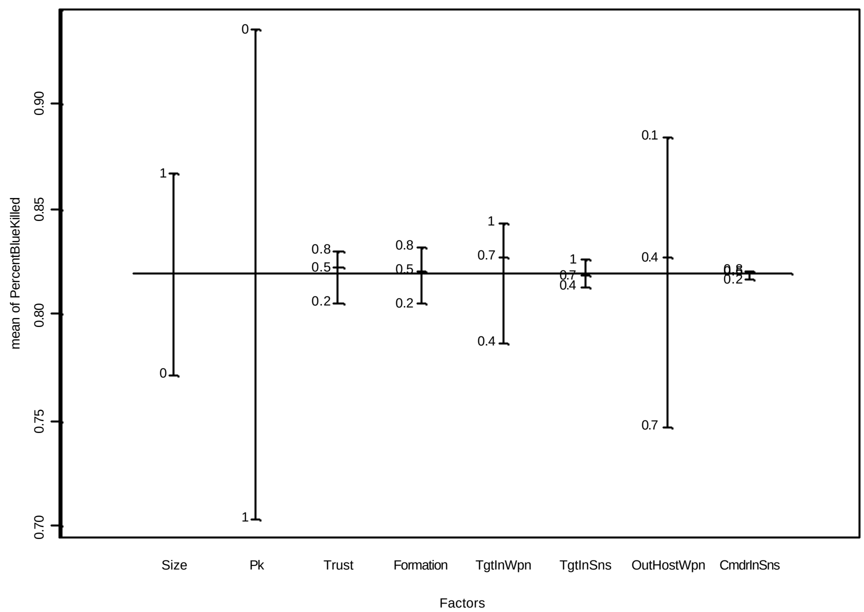

Figure 17. Plot of Treatment Means for PercentBlueKilled from the Factorial Design

The boxplot of the factors is next examined, as shown in Figures 18. In addition to the same indications as the mean treatment plot shows, the boxplot also displays the variability of the data. It can be seen that for a large total force size and low Pk, there are a large number of outliers. As a result of the numerous low outliers, coupled with the lack of high outliers, as the percentage of Blue killed is bounded by 1.0, we will expect to see a negatively skewed distribution of data and treatment means that are generally below the respective medians.

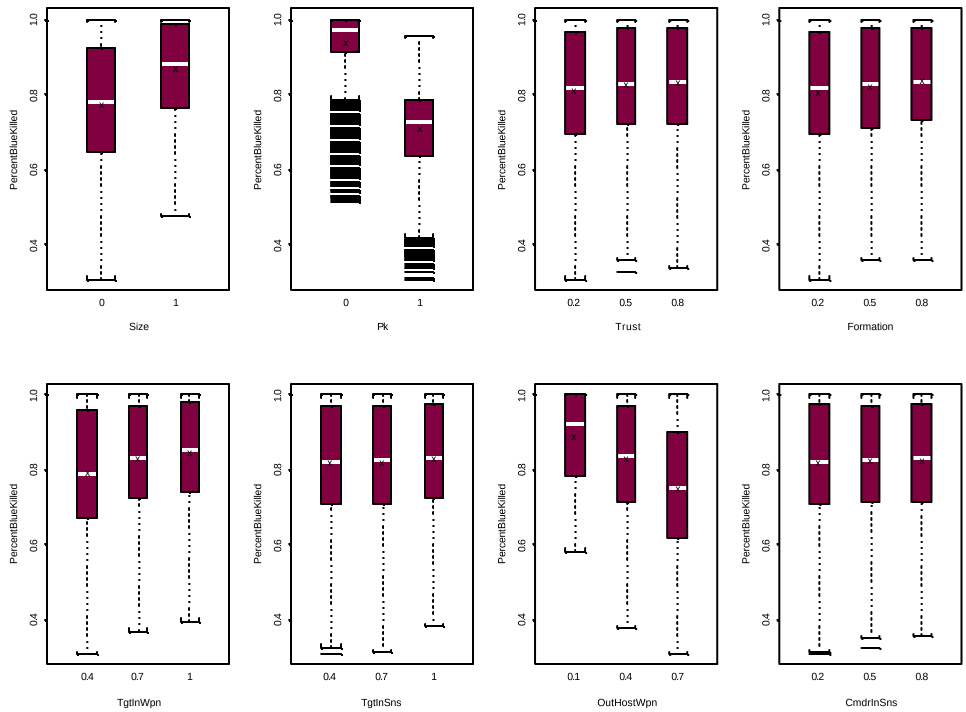

Figure 18. Boxplot of Factors for PercentBlueKilled from the Factorial Design

Figures 19 and 20 shows the qq-plot and the residual plot of the PercentBlueKilled model respectively. Unlike the Latin Hypercube Design, no transformation is required in the Factorial Design. The two plots show that the residuals are normally distributed and non-homoscedastic. However, the residual plot demonstrates a straight-line boundary at the upper right corner, which is a result of having the percentage of Blue killed equals 1, mostly occurring at low Pk. Despite some unequal variances, we can still use the ANOVA technique to determine terms that have significant effects.

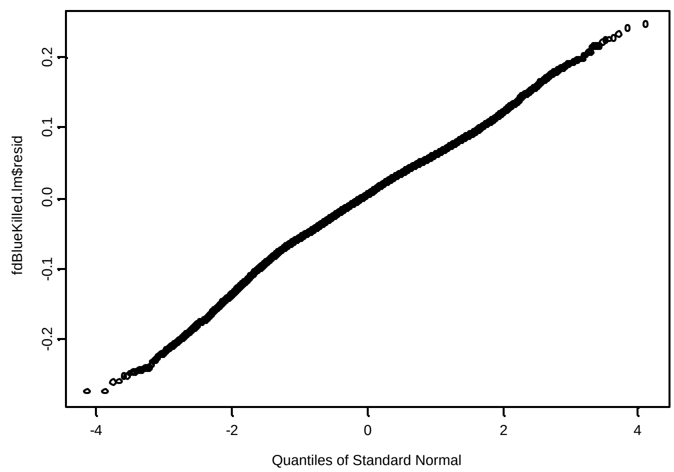

Figure 19. QQ-plot of the PercentBlueKilled Model from the Factorial Design

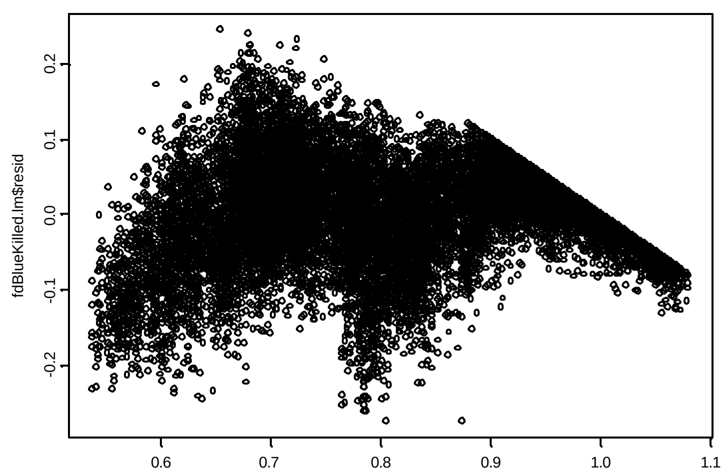

Figure 20. Residual Plot of the PercentBlueKilled Model from the Factorial Design

The ANOVA result for the PercentBlueKilled is shown in Table 9 below. The very first observation that one makes is the 56 percent of the total sum of squares that the Pk accounts for. The Size, TgtInWpn and OutHostWpn are the other three factors that are statistically significant and account for the large sum of squares. Although the remaining four main effects are also statistically significant, their small sum of squares justify their exclusion from the model. The above four main effects, together with the

Size:OutHostWpn and the TgtInWpn:OutHostWpn, the two significant interacting terms that have their sum of squares greater than 1 (all shown in bold in Table 9), can account for 85.95 percent of the total sum of squares. This model is certainly a much simpler model, losing only about 3.9 percent of information on the variability of the data.

ANOVA TABLE FOR PERCENTBLUEKILLED

Terms	Df	Sum of Sq	Mean Sq	F value	Pr(F)	%Sum of Sq
Size	**1**	**39.1286**	**39.1286**	**16120.56**	**0.0000**	**9.4146**
Pk	**1**	**232.8432**	**232.8432**	**95928.8**	**0.0000**	**56.0234**
Trust	2	1.7764	0.8882	365.93	0.0000	0.4274
Formation	2	2.1229	1.0614	437.3	0.0000	0.5108
TgtInWpn	**2**	**9.8187**	**4.9093**	**2022.59**	**0.0000**	**2.3624**
TgtInSns	2	0.4819	0.241	99.27	0.0000	0.1159
OutHostWpn	**2**	**55.0833**	**27.5417**	**11346.86**	**0.0000**	**13.2534**
CmdrInSns	2	0.0632	0.0316	13.03	0.0000	0.0152
Size:Pk	1	0.589	0.589	242.67	0.0000	0.1417
Size:Trust	2	0.5659	0.283	116.58	0.0000	0.1362
Size:Formation	2	0.4378	0.2189	90.18	0.0000	0.1053
Size:TgtInWpn	2	1.6039	0.802	330.4	0.0000	0.3859
Size:TgtInSns	2	0.3031	0.1516	62.44	0.0000	0.0729
Size:OutHostWpn	**2**	**13.8327**	**6.9164**	**2849.47**	**0.0000**	**3.3282**
Size:CmdrInSns	2	0.0653	0.0326	13.44	0.0000	0.0157
Pk:Trust	2	0.0171	0.0086	3.53	0.0294	0.0041
Pk:Formation	2	0.0042	0.0021	0.87	0.4197	0.0010
Pk:TgtInWpn	2	0.007	0.0035	1.44	0.2374	0.0017
Pk:TgtInSns	2	0.0205	0.0103	4.22	0.0147	0.0049
Pk:OutHostWpn	2	1.249	0.6245	257.28	0.0000	0.3005
Pk:CmdrInSns	2	0.0328	0.0164	6.76	0.0012	0.0079
Trust:Formation	4	0.5398	0.135	55.6	0.0000	0.1299
Trust:TgtInWpn	4	0.6049	0.1512	62.3	0.0000	0.1455
Trust:TgtInSns	4	0.1125	0.0281	11.58	0.0000	0.0271
Trust:OutHostWpn	4	2.2263	0.5566	229.3	0.0000	0.5357
Trust:CmdrInSns	4	0.0016	0.0004	0.17	0.9555	0.0004
Formation:TgtInWpn	4	0.7061	0.1765	72.73	0.0000	0.1699
Formation:TgtInSns	4	0.0671	0.0168	6.91	0.0000	0.0161
Formation:OutHostWpn	4	2.3769	0.5942	244.82	0	0.5719
Formation:CmdrInSns	4	0.0054	0.0014	0.56	0.6945	0.0013
TgtInWpn:TgtInSns	4	0.0187	0.0047	1.93	0.1029	0.0045
TgtInWpn:OutHostWpn	**4**	**6.5057**	**1.6264**	**670.07**	**0.0000**	**1.5653**
TgtInWpn:CmdrInSns	4	0.0069	0.0017	0.71	0.5866	0.0017
TgtInSns:OutHostWpn	4	0.1428	0.0357	14.71	0.0000	0.0344
TgtInSns:CmdrInSns	4	0.0152	0.0038	1.57	0.1806	0.0037
OutHostWpn:CmdrInSns	4	0.0169	0.0042	1.74	0.138	0.0041
Residuals	17396	42.2244	0.0024			10.1594

Total Sum of Sq	415.6177
Multiple R-Squared	0.8984

Table 9. ANOVA Table for PercentBlueKilled

3. Comparison Between Latin Hypercube and Factorial Design

A comparison is made between the terms that have significant effects from the two designs for MOE 1. It can be seen from Table 10 below that the Size and Pk, the two terms with the greatest sum of squares, are common in both models. The only terms with a large sum of squares that are selected in ANOVA but not in AIC model are the Size:OutHostWpn and TgtInWpn:OutHostWpn. Based on the simplified AIC mode1 (see Table 13), the multiple R-squared value that can be achieved is 79.56 percent, if a Full Factorial design were to be carried out. Thus, the Latin Hypercube Design is able to fit a model considerably well for MOE 1, with much fewer runs (4800 runs as compared to 174,960 runs in Factorial Design.)

Terms	Latin Hypercube Design	Factorial Design	%Sum of Square
Size	*	*	9.4146
Pk	*	*	56.0234
Formation	*		0.5108
TgtInWpn		*	2.3624
TgtInSns	*		0.1159
OutHostWpn	*	*	13.2534
Size:Pk	*		0.1417
Size:Formation	*		0.1053
Size:OutHostWpn		*	3.3282
TgtInWpn:OutHostWpn		*	1.5653

Table 10. Comparison of the PercentBlueKilled Model in Latin Hypercube And Factorial Design

B. MOE 2: FRACTIONAL EXCHANGE RATIO (FER)

1. Latin Hypercube Design

Figures 21 and 22 show the qq-plot and the residual plot of the regression model respectively, obtained by applying a logarithmic transformation to FER. The qq-plot shows that the residuals are normally distributed except at the tail ends. The residual plot shows two distinct blocks, one block belonging to the group of data with the low Pk, and the other block belonging to the group with the high Pk. To enlarge the residuals, the residuals of one of the blocks (low Pk) are shown in Figure 23. It can be seen that the residuals are not homoscedastic.

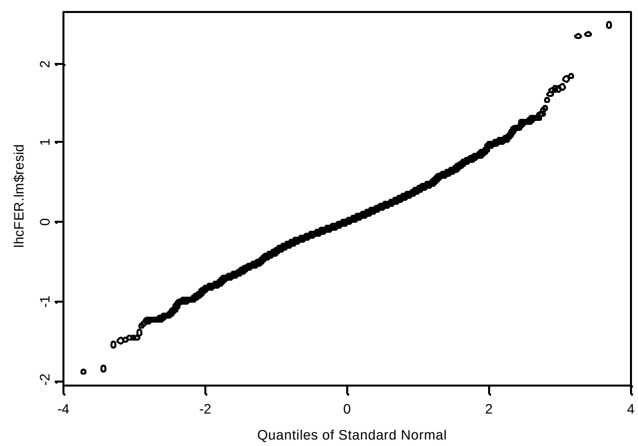

Figure 21. QQ-plot of the FER Model from the Latin Hypercube Design

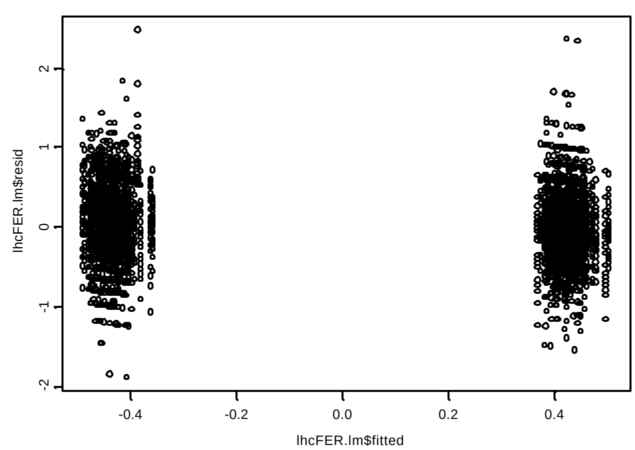

Figure 22. Residual Plot of the FER Model from the Latin Hypercube Design

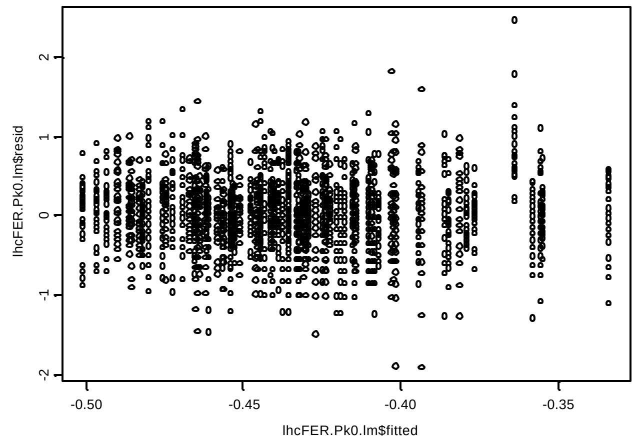

Figure 23. Residual Plot of the FER Model at Low Pk from the Latin Hypercube Design

The stepAIC function is again used to find the model with the lowest AIC value. Table 11 lists the terms chosen. It is shown that Size, Pk, and Formation are statistically significant. Of all the 11 interacting terms listed, only eight are statistically significant (marked with an asterisk). Once again, if we include just the significant main effects and the interacting terms that contain only the significant main effects as shown in bold, we obtain a simpler model that has five terms, one that has a multiple R-Squared value of 0.5008. This again is negligibly differently from the full AIC model that has a multiple R-Squared value of 0.507.

The positive coefficients of the significant main effects, such as the Pk and the Formation, show that the FER increases as the values of these factors increase, except for the Size. In addition, the coefficient of the Pk is shown to be more than two times that of the Formation, which will yield a higher return in FER. The Size again has a negative coefficient, which is in line with the initial observation on MOE 1 in the earlier model.

Coefficients:

	Value	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.2659	0.1438	-1.8493	0.0645
Size	**-0.1326**	**0.0619**	**-2.1435**	**0.0321**
Pk	**0.7817**	**0.0296**	**26.3924**	**0.0000**
Trust	-0.0529	0.0846	-0.6253	0.5318
Formation	**0.3362**	**0.1048**	**3.2079**	**0.0013**
TgtInWpn	-0.3240	0.1748	-1.8535	0.0639
TgtInSns	-0.2068	0.1911	-1.0821	0.2793

CmdrInSns	0.0591	0.1214	0.4869	0.6263
OutHostWpn	0.0916	0.1275	0.7185	0.4725
Formation:TgtInWpn	-0.4555	0.1388	-3.2809	0.0010*
Size:TgtInWpn	0.1785	0.0682	2.6177	0.0089*
Size:Formation	**-0.1066**	**0.0452**	**-2.3566**	**0.0185***
Pk:CmdrInSns	0.0982	0.0454	2.1661	0.0304*
Size:Pk	**0.0516**	**0.0247**	**2.0908**	**0.0366***
TgtInWpn:TgtInSns	0.5073	0.2331	2.1759	0.0296*
Size:Trust	0.0815	0.0453	1.7991	0.0721
TgtInSns:CmdrInSns	-0.5145	0.1865	-2.7589	0.0058*
Trust:CmdrInSns	0.3581	0.1319	2.7154	0.0066*
Trust:OutHostWpn	-0.2611	0.1425	-1.8323	0.0670
Formation:OutHostWpn	0.2331	0.1636	1.4250	0.1542

```
Residual standard error: 0.4277 on 4780 degrees of freedom
Multiple R-Squared: 0.507
F-statistic: 258.8 on 19 and 4780 degrees of freedom, the p-value is 0
```

Table 11. AIC Model for the FER from the Latin Hypercube Design

2. Factorial Design

Similar to the PercentBlueKilled, the data for the FER is first examined graphically using S-Plus functionality plots. Figure 24 shows the plot of the treatment means of the factors. The effects of Size, Pk, and OutHostWpn clearly dominate the others, which show almost negligible effects.

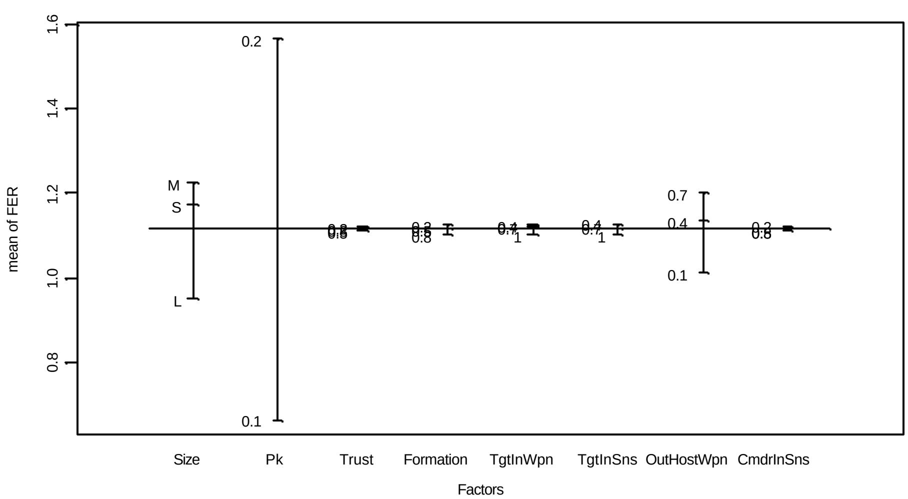

Figure 24. Plot of Treatment Means for FER from the Factorial Design

The boxplot in Figure 25 below shows similar indications as the treatment mean plot, that is, the Size, Pk, and OutHostWpn have greater effects than the other factors have. In addition, the boxplots show that there are far more high outliers than low outliers, which means we will expect to see a more negatively skewed distribution.

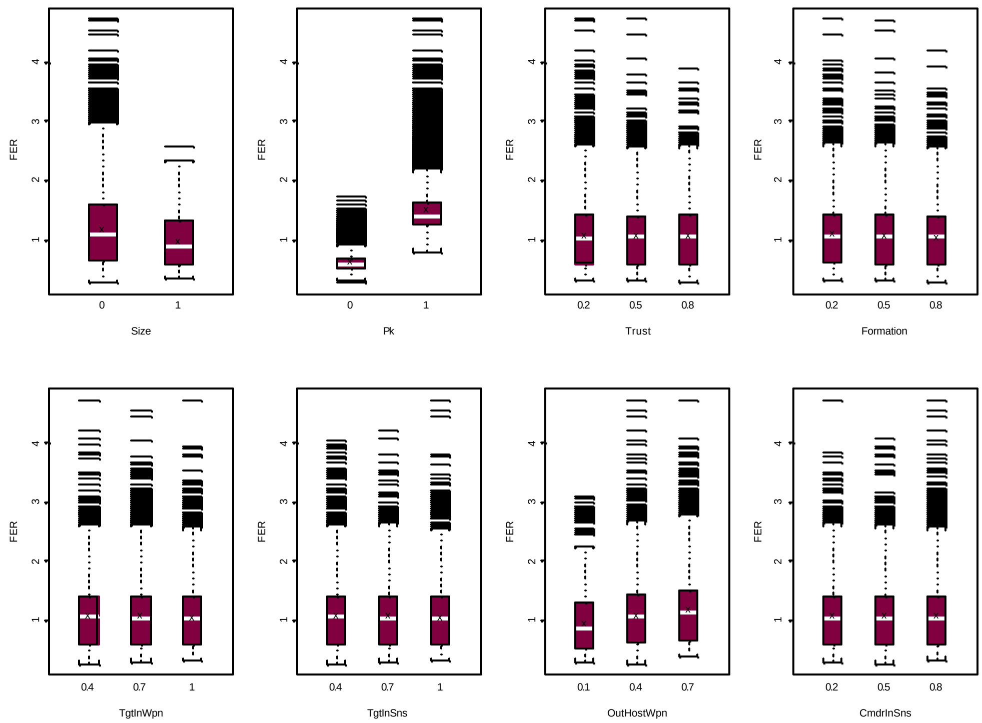

Figure 25. Boxplot of Factors for the FER from the Factorial Design

For the FER, a transformation of log(FER + 0.1) is applied to the responses. The qq-plot and the residual plot are shown in Figures 26 and 27 respectively. The linear trend in the qq-plot indicates normally distributed errors and the residual plot shows non-homoscedasticity. As can be seen from the residual plot, similar to the residual plot of the FER in the Latin Hypercube Design, the residuals are divided into two blocks, each block corresponds to the responses of each Pk. In addition, there are strips of residual grouping visually, each of which might correspond to particular set treatments. Infering that a pattern—relating to one of the sub-objectives of the thesis to map the outputs to the inputs to derive possible patterns—might exist, a detailed study of the data was conducted. However, only the two groups of data circled in green are distinctly different from the others, as shown in the text boxes. Those data in the red box are a good mix, except that they are distinguishable by their Pks.

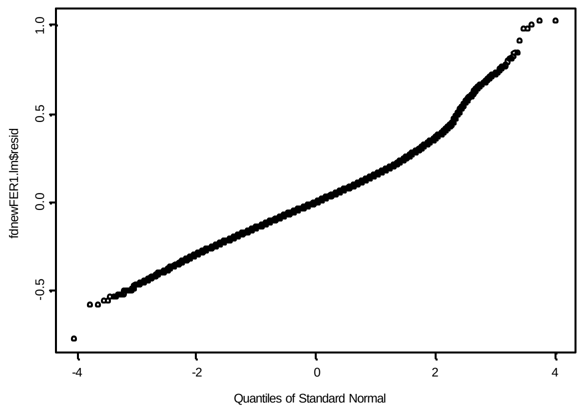

Figure 26. QQ-plot of the FER Model from the Factorial Design

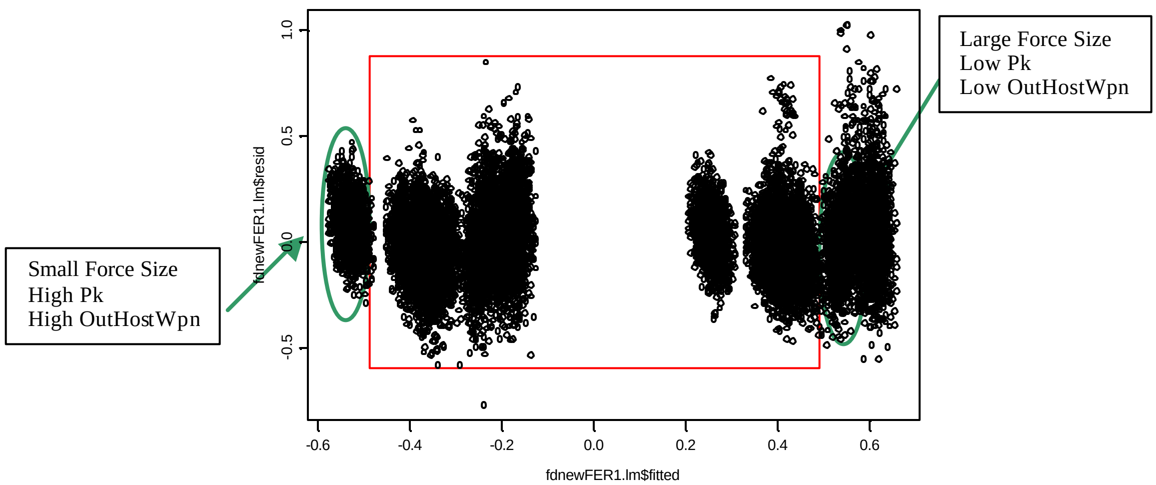

Figure 27. Residual Plot of the FER Model from the Factorial Design

The ANOVA Table for the FER is shown in Table 12. It is observed that the Size, Pk, and OutHostWpn have the most significant main effects, with Pk being the most dominating. Its sum of squares alone composed almost 79.5 percent of the total sum of squares. These three terms account for 86.07 percent of the total sum of squares and can describe the data sufficiently, since the sum of squares for the remaining main effects are negligibly small. Although there are other interacting effects, their sum of squares are also small as compared to the three significant main effects.

ANOVA TABLE FOR FER

Terms	Df	Sum of Sq	Mean Sq	F value	Pr(F)	%Sum of Sq
Size	**1**	**114.867**	**114.867**	**5109.8**	**0**	**3.3927**
Pk	**1**	**2690.367**	**2690.367**	**119679.8**	**0.0000**	**79.4621**
Trust	2	0.759	0.379	16.9	0.0000	0.0224

Formation	2	1.684	0.842	37.4	0.0000	0.0497
TgtInWpn	2	1.673	0.836	37.2	0.0000	0.0494
TgtInSns	2	2.311	1.155	51.4	0.0000	0.0683
OutHostWpn	**2**	**109.019**	**54.51**	**2424.8**	**0.0000**	**3.2200**
CmdrInSns	2	0.13	0.065	2.9	0.0560	0.0038
Size:Pk	1	4.061	4.061	180.7	0.0000	0.1199
Size:Trust	2	5.646	2.823	125.6	0.0000	0.1668
Size:Formation	2	3.174	1.587	70.6	0.0000	0.0937
Size:TgtInWpn	2	0.297	0.148	6.6	0.0014	0.0088
Size:TgtInSns	2	1.113	0.556	24.7	0.0000	0.0329
Size:OutHostWpn	2	29.332	14.666	652.4	0.0000	0.8663
Size:CmdrInSns	2	0.98	0.49	21.8	0.0000	0.0289
Pk:Trust	2	0.002	0.001	0	0.9626	0.0001
Pk:Formation	2	0.044	0.022	1	0.3729	0.0013
Pk:TgtInWpn	2	0.299	0.15	6.7	0.0013	0.0088
Pk:TgtInSns	2	0.119	0.059	2.6	0.0711	0.0035
Pk:OutHostWpn	2	0.768	0.384	17.1	0.0000	0.0227
Pk:CmdrInSns	2	0.064	0.032	1.4	0.2407	0.0019
Trust:Formation	4	2.673	0.668	29.7	0.0000	0.0789
Trust:TgtInWpn	4	0.64	0.16	7.1	0.0000	0.0189
Trust:TgtInSns	4	0.251	0.063	2.8	0.0246	0.0074
Trust:OutHostWpn	4	6.435	1.609	71.6	0.0000	0.1901
Trust:CmdrInSns	4	0.027	0.007	0.3	0.8793	0.0008
Formation:TgtInWpn	4	0.901	0.225	10	0.0000	0.0266
Formation:TgtInSns	4	0.135	0.034	1.5	0.2004	0.0040
Formation:OutHostWpn	4	7.423	1.856	82.6	0	0.2192
Formation:CmdrInSns	4	0.067	0.017	0.8	0.5577	0.0020
TgtInWpn:TgtInSns	4	0.42	0.105	4.7	0.0009	0.0124
TgtInWpn:OutHostWpn	4	7.824	1.956	87	0.0000	0.2311
TgtInWpn:CmdrInSns	4	0.109	0.027	1.2	0.3039	0.0032
TgtInSns:OutHostWpn	4	0.66	0.165	7.3	0.0000	0.0195
TgtInSns:CmdrInSns	4	0.121	0.03	1.3	0.2509	0.0036
OutHostWpn:CmdrInSns	4	0.273	0.068	3	0.0161984	0.0081
Residuals	17396	391.057	0.022			11.5502
Total Sum of Sq		3385.725				
Multiple R-Squared	0.8845					

Table 12. ANOVA Table for the FER from the Factorial Design

3. Comparison Between Latin Hypercube and Factorial Design

A comparison is made between the terms that have significant effects from the two designs for MOE 2, as shown in Table 13 below. The simplified AIC model will achieve a multiple R-Squared value of 0.8312, if a full Factorial Design were to be run.

Again, the Latin Hypercube Design has shown that it is able to fit a model considerably well using far fewer runs.

Terms	Latin Hypercube Design	Factorial Design	%Sum of Squares
Size	*	*	3.3927
Pk	*	*	79.4621
Formation	*		0.0497
OutHostWpn		*	3.2200
Size:Pk	*		0.1199
Size:Formation	*		0.0937

Table 13. Comparison of the FER Model in Latin Hypercube and Factorial Designs

C. COMPARISON BETWEEN VIRGINIA AND MHPCC'S OUTPUTS

As mentioned in Chapter III, a set of data for the small force size scenario was collected at Virginia Cluster (VC) as a backup due to the initial data reading problem at MHPCC. The data were used together with the output of the large force size scenario from the MHPCC in the initial analysis. However, when some irregularities occurred in the residual plot of the linear model obtained, additional data for the small force size scenario was generated at MHPCC, and the output was compared with the output from the Virginia Cluster. Using only the data for the small force size, a linear model was fit on the data from both the Virginia and the MHPCC, with each of the residual plots shown in Figures 28 and 29 respectively. The two figures clearly show data of a different nature. Subsequently, two methods were used to compare the outputs: a two-sample t-test and an ANOVA.

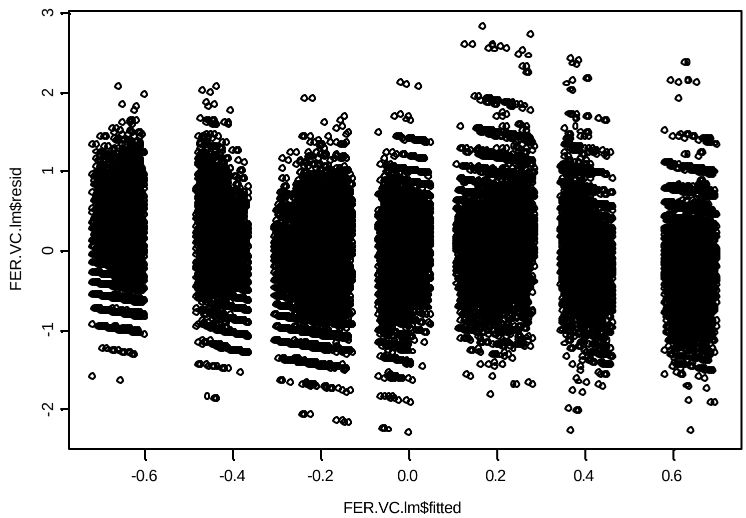

Figure 28. Residual Plot of the FER Model for Small Force Size Scenario from Virginia Cluster's Output

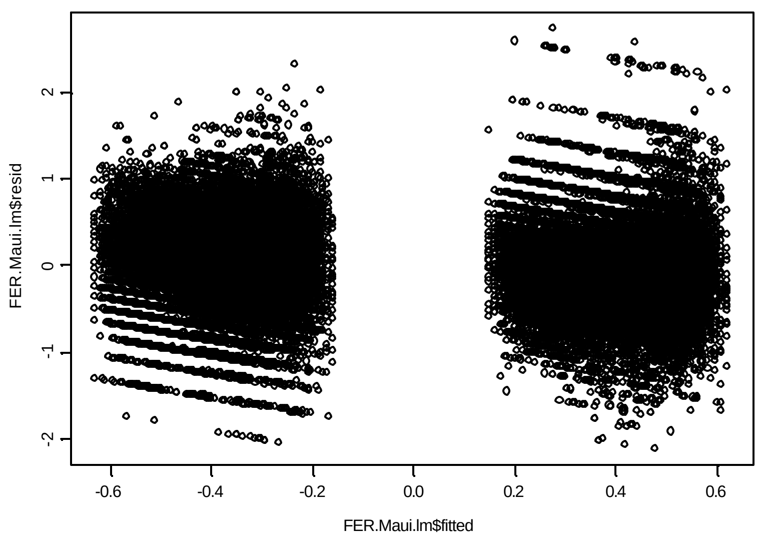

Figure 29. Residual Plot of the FER Model for Small Force Size Scenario from MHPCC's Output

1. Two-sample T-Test

Tables 14 and 15 below show the results of the two-sample t-test for the PercentBlueKilled and the FER using the raw data, both tests assuming equal variances. The use of the raw data will produce a more accurate result. The t-tests concluded that while there is no statistical difference between the Virginia and MHPCC's outputs for

FER, there is a statistically significant difference between the two outputs for PercentBlueKilled. However, the difference between 0.78 and 0.774 is not practically significant.

t-Test for PercentBlueKilled: Two-Sample Assuming Equal Variances

	PercentBlueKilled_MHPCC	*PercentBlueKilled_VC*
Mean	0.779968278	0.774154092
Variance	0.05164929	0.0535291
Observations	43740	43740
Pooled Variance	0.052589195	
Hypothesized Mean Difference	0	
df	87478	
t Stat	3.749428337	
P(T<=t) one-tail	8.86761E-05	
t Critical one-tail	1.64487119	
P(T<=t) two-tail	0.000177352	
t Critical two-tail	1.959992915	

Since t Stat of 3.749>1.96, reject H_0, i.e., there is a difference between the two outputs.

Table 14. Result of T-test for PercentBlueKilled

t-Test: Two-Sample Assuming Equal Variances

	FER_MHPCC	*FER_VC*
Mean	1.137853556	1.141652869
Variance	1.000017544	0.926914238
Observations	43740	43740
Pooled Variance	0.963465891	
Hypothesized Mean Difference	0	
df	87478	
t Stat	-0.572415416	
P(T<=t) one-tail	0.283521023	
t Critical one-tail	1.64487119	
P(T<=t) two-tail	0.567042045	
t Critical two-tail	1.959992915	

Since t Stat of -0.572>-1.96, accept H_0, i.e., there is no difference between the two outputs.

Table 15. Result of T-test for FER

2. ANOVA

Table 16 below compares the outputs of the Virginia Cluster and the MHPCC, for both MOEs. The main observation made is that the effect of Pk is insignificant in the

Virginia's output, whereas the effect of Pk in the MHPCC's output is consistent with the results shown earlier in the Latin Hypercube and Factorial Designs, that not only is the effect of Pk always dominating the effects of the other factors, it dominates by a large amount. The cause of the insignificant effect of Pk in the Virginia's output remains undetermined.

		PercentBlueKilled				FER			
		VC		MHPCC		VC		MHPCC	
Terms	**Df**	**Sum of Sq**	**Pr(F)**	**Sum of Sq**	**Pr(F)**	**Sum of Sq**	**Pr(F)**	**Sum of Sq**	**Pr(F)**
Pk	1	0.002	0.8655655	1172.653	0	0.05	0.674379	6616.15	0
Trust	2	840.26	0	27.787	0	4971.71	0	39.46	0
Formation	2	613.721	0	32.581	0	1607.72	0	42.09	0
TgtInWpn	2	2.588	0	98.521	0	37.64	0	4.84	0.000098
TgtInSns	2	16.072	0	7.064	0	1.41	0.0715074	11.7	0
OutHostWpn	2	3.042	0	645.639	0	2.3	0.0136822	406.19	0
CmdrInSns	2	5.491	0	0.54	0.0063171	4	0.0005765	6.43	0.0000047
Pk:Trust	2	0.075	0.5215888	0.017	0.8501824	0.82	0.2172026	0.28	0.5883674
Pk:Formation	2	0.298	0.0756615	0.017	0.8527861	1.55	0.0549331	0.38	0.4841551
Pk:TgtInWpn	2	0.122	0.3479416	4.504	0	0.24	0.636	3.94	0.0005405
Pk:TgtInSns	2	0.672	0.0029305	0.131	0.2929654	0.59	0.3342848	0.78	0.2251958
Pk:OutHostWpn	2	2.809	0	3.64	0	6.35	0.0000071	6.19	0.0000074
Pk:CmdrInSns	2	14.82	0	0.398	0.0239753	7.95	0.0000004	0.66	0.285562
Trust:Formation	4	188.952	0	1.12	0.0003186	1095.52	0	1.57	0.1997013
Trust:TgtInWpn	4	12.906	0	0.363	0.1461572	119.91	0	2.3	0.067162
Trust:TgtInSns	4	3.047	0	0.424	0.0933932	16.33	0	0.41	0.8172562
Trust:OutHostWpn	4	0.054	0.9185402	11.924	0	0.33	0.8754123	13.81	0
Trust:CmdrInSns	4	0.071	0.8733332	0.035	0.9574468	0.27	0.9079565	0.39	0.8316998
Formation:TgtInWpn	4	233.7	0	0.225	0.3777403	107.54	0	3.27	0.0142042
Formation:TgtInSns	4	30.912	0	0.472	0.0652118	14.08	0	0.18	0.9506168
Formation:OutHostWpn	4	3.848	0	11.573	0	0.99	0.4500256	16.16	0
Formation:CmdrInSns	4	0.701	0.0161561	0.097	0.7678432	2.7	0.038944	0.2	0.9442865
TgtInWpn:TgtInSns	4	8.771	0	1.2	0.00016	38.83	0	7.73	0.0000062
TgtInWpn:OutHostWpn	4	1.156	0.0004874	54.06	0	11.31	0	29.28	0
TgtInWpn:CmdrInSn s	4	0.198	0.4869222	0.269	0.2830576	0.76	0.5834155	0.57	0.7056275
TgtInSns:OutHostWpn	4	58.203	0	1.681	0.0000024	9.24	0.0000006	3.37	0.0118844
TgtInSns:CmdrInSns	4	6.252	0	0.493	0.0551691	1.24	0.329003	1.75	0.1544071
OutHostWpn:CmdrInSns	4	10.632	0	0.742	0.007589	5.82	0.0002281	0.52	0.7394914
Total	43654	2515.914		2328.647		11693.36		11437.16	

Table 16. Comparison of Significant Effects Between Outputs of Virginia Cluster and MHPCC for Both MOEs

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS

The ongoing revolution in military affairs is transforming military warfare. Modern combat systems are increasingly more effective and complex to operate. Nonetheless, their complexities cannot be compared to the complexity of human behaviors. SOCRATES, like other agent-based models developed under Project Albert, attempts to capture some of these behaviors. The intent of this thesis was to study, through SOCRATES, how human factors affect combat outcomes and to gain some answers to the questions stated in Chapter I. The primary findings are now summarized.

A. RANGE OF PARAMETERS TO BE USED

It has been shown that it is important for the appropriate range of parameters to be used in the inputs to obtain intuitive outcomes. This applies not only to both the value components of the movement decision component and the Pk, it also applies to the tactic0 and tactic1 decision components, which were not examined in this thesis. The thesis has also shown that the recommended values for the value components of the movement decision tabulated in Table 4, as well as values of the Pk between 0.1 and 0.4, should be used when modeling a scenario of similar nature. The value of Pk used is especially important as its effect was shown to be highly significant, generally accounting for more than 50 percent of the total sum of squares.

B. COMPARISON BETWEEN LATIN HYPERCUBE AND FACTORIAL DESIGNS

The outcomes on the significant effects obtained from both the Latin Hypercube Design and the Factorial Design are summarized in Table 17 below. The multiple R-Squared value is the value that would be obtained if the model shown were used to fit data from a full Factorial Design of 2 scenarios x 2 Pks x 3 levels for the six value components of the movement decision. The thesis has shown that the Latin Hypercube Design can provide sufficient data for the user to determine factors that have important effects and obtain a model that can fit the data considerably well with much fewer runs than a Factorial Design requires.

Terms	MOE 1: PercentBlueKilled		MOE 2: FER	
	Latin Hypercube	Factorial	Latin Hypercube	Factorial
Size	*	*	*	*
Pk	*	*	*	*
Formation	*		*	
TgtInWpn		*		
TgtInSns	*			
OutHostWpn	*	*	*	*
Size:Pk	*			
Size:Formation	*		*	
Size:OutHostWpn		*	*	
TgtInWpn:OutHostWpn		*		
Multiple R-Squared	0.7956	0.8595	0.8312	0.8607

Table 17. Summary of Comparison of Significant Effects for the Latin Hypercube and Factorial Designs

C. RELATIONSHIPS BETWEEN SIZE, PK, FORMATION AND OUTHOSTWPN

From Table 17, since the factors Size, Pk, Formation, and OutHostWpn are shown to be significant in all of the models, it may be worthwhile to look at the relationship between each of them. This was done by using the Vistool, a feature of SOCRATES that allows the user to view the inter-relationships between any three factors at a time. Figures 30 and 31 show the three-dimensional plot of Pk, Formation, and OutHostWpn for the small and large force size respectively. The Pk and the OutHostWpn factors are plotted along the two horizontal axes, and the third factor of Formation being varied using the sliding scale on the left. The vertical axis displays the number of Blue (indirectly the percentage of Blue killed) and Red killed, each represented by the Blue and the Red colored planes respectively. The following are noted:

a. When the Pk is low: Both figures show that the majority of the Blue are killed.

b. When the formation factor equals 0.2 and when Pk is high: In the small force size scenario, the number of Blue killed is the lowest when the

OutHostWpn is ***low***. The same result is obtained for mid and high levels of the Formation factor. However, when the force size is large, the lowest percentage of Blue killed occurs at the mid value of the OutHostWpn. It is also seen (result not displayed) that the lowest percentage of Blue killed occurs at either the ***mid or high*** level of OutHostWpn regardless of the levels of the Formation factor.

This observation suggests that when the overall force size is small, and if the Blue is more capable, it will be to Blue's advantage to be more aggressive in order to optimize its superior capability. However, when the total force size is large, being overly aggressive may lead to a higher casualty rate for Blue than if it is not overly aggressive, despite its superior capability. This phenomenon might be attributed to the reason that when the overall force size is large, that after each exchange of fire there are more Red survivors who can return fire. The higher percentage of Blue killed when the total force size is large (as shown in Figure 17) can thus be explained. This may also indicate the existence of some emergent behaviors that only appear at a larger force size.

D. HOW CAPABLE MUST BLUE BE?

Although Figures 30 and 31 do not show the full range of the factors, they provide sufficient indication that by being twice as lethal as the Red, the Blue can achieve a higher FER (relative comparison of the gaps between the number of Blue and Red killed at low and high Pk). It can be seen that at high Pk, the percentage of Blue killed, or the number of Blue killed, can be reduced by about 50 percent or more (there are a total of 16 and 28 Blue agents in the small and large force size scenarios respectively.) As it has been suggested, some emergent patterns may exist. The Pk may need to be increased disproportionately to achieve the same reduced attrition rate, that is, 50 percent or less, if the total force size is increased. Of course, this required increment in the Pk may vary with the scenario.

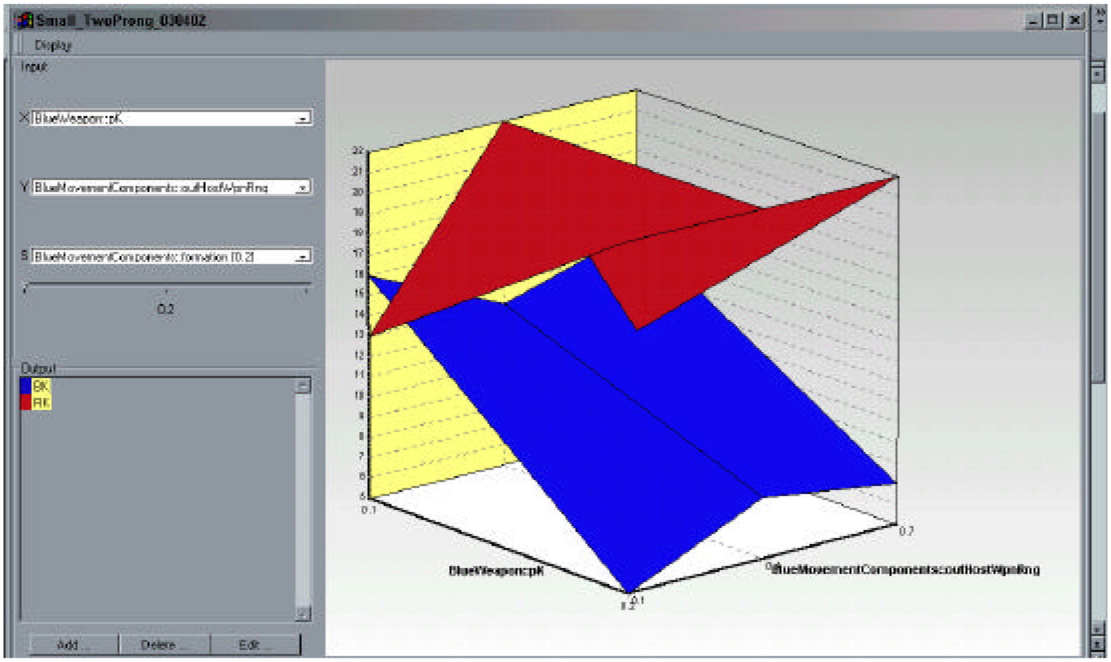

Figure 30. Three-dimensional Plot of the Relationships Between the Pk, Formation (=0.2) and OutHostWpn for Small Total Force Size

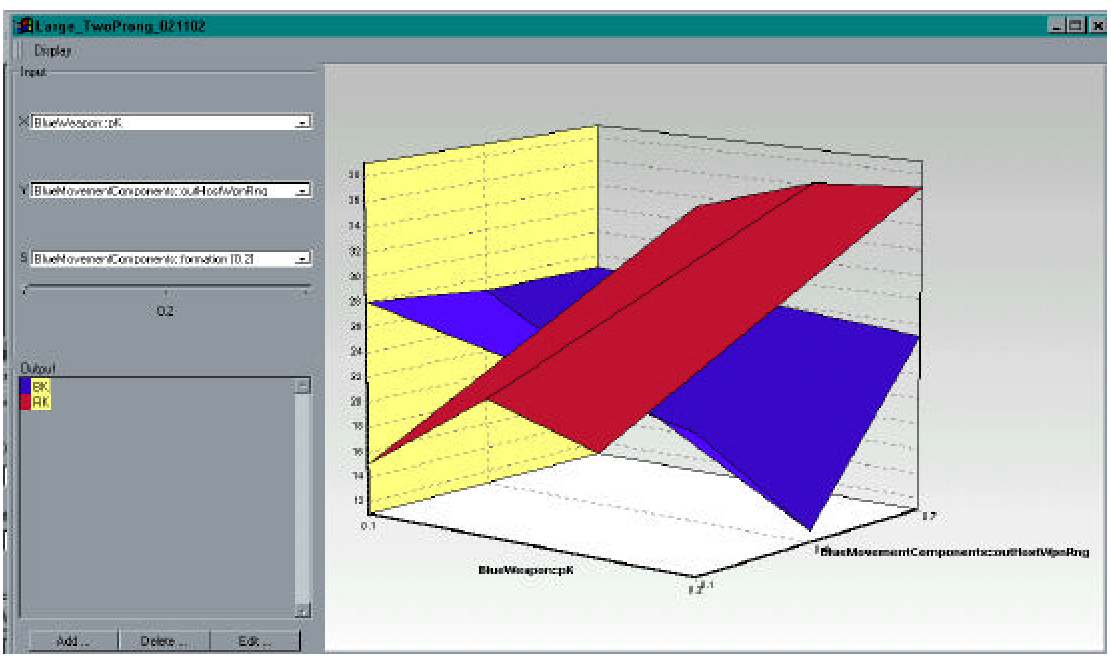

Figure 31. Three-dimensional Plot of the Relationships Between the Pk, Formation (=0.2) and OutHostWpn for Large Total Force Size

E. DIFFERENCE BETWEEN MHPCC AND VIRGINIA CLUSTER'S OUTPUTS

The t-tests and the ANOVA in Chapter IV show that there are possible differences between the outputs of MHPCC and Virginia Cluster on the same scenario. In the output of Virginia Cluster, the Pk appeared to be an insignificant factor, which is contrary to the conclusion from the MHPCC's output. The reason for the distinct difference in the importance of Pk cannot be ascertained in this thesis.

F. SOCRATES EVALUATION

In addition to the above-mentioned findings, the thesis has also served its objective of evaluating SOCRATES to help refine it. The following problems were uncovered:

a. The weapon of a killed agent continues to fire.

b. The obstruction inhibits only the movements of the agents, but does not inhibit firing.

c. Whenever there is a tie between some of the movement alternatives, the first one considered will always be selected for the agent, hence producing a biased decision.

d. There was a data-reading problem in Maui's configuration when it upgraded its version to 2.2.1, which resulted in repetitive outputs, as the inputs were unable to be read in. The problem was discovered and rectified.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. RECOMMENDATIONS

The following is a list of recommendations for SOCRATES for possible future research:

a. SOCRATES' Development:

i. **Employment of Weapon:** The priority in the employment of weapon in the present version of SOCRATES is based on the Pk, rather than on the effectiveness. If an agent is equipped with two different types of weapons, the weapon with the highest Pk will be selected, regardless of the type of target he is engaging. This will result in unsound weapon employment if more than one type of weapon is modeled in a scenario. For example, an agent equipped with an anti-tank guided missile weapon system and an M16 will shoot an enemy on foot with the anti-tank weapon system rather than the M16, since the Pk of the anti-tank guided missile weapon system is much higher than the M16. A provision should be made to tag a type of weapon to specific targets so that the agents can only select a weapon with the highest Pk among a list of weapons appropriate for that target.

ii. **Selection of Pk:** The Pk used in this thesis is derived from the results of the preliminary runs. The value of the Pk is selected so that the runs produce intuitive outcomes in the scenario modeled, instead of basing the value on the engineering data of the actual weapon system. This will result in a subjective Pk being selected, and may affect the validation of SOCRATES in future.

iii. **Bias in Alternative Selection in Decision-Making:** Currently, there is a bias in the selection of the best alternative in the movement decision. Whenever there is a tie in the scores between some of the alternatives, the first best alternative being considered will always be selected to be the best decision, and the sequence of consideration is always the same. To minimize this bias, a modification can be made so that either an alternative is selected randomly, or the one that best meets the overall mission requirement among the tied alternatives, is chosen as the best decision.

iv. **Fire Obstruction:** The present obstacles in SOCRATES only inhibit movement. This feature can be modified to include fires so that urban warfare can be modeled in SOCRATES.

b. Experimental Design:

i. **Full Factorial Design:** In the Factorial Design, the Size and Pk have only two levels, while the remaining six factors have three levels. This unbalanced number of levels of factors makes the breakdown of the sum of squares due to the possible linear and quadratic effects of each term more difficult. In a follow-on study, additional runs can be made using another force size (median) and Pk (0.15). Combining this set of data with those already generated will make the design a full 3^8 factorial design. Then, Yate's algorithm [Ref 12] can be used to breakdown the sum of squares for the exploration of the polynomial effects of each term.

c. Analysis:

i. **Exploration of Polynomial Effects:** As mentioned in b(i) above, the polynomial effects of each term can be explored using the Yate's algorithm to breakdown the sum of square.

ii. **Significant Effect of Variables on MOEs:** Tukey's procedure can be used to simultaneously test all pair-wise means for significant differences with a specified overall type I error rate. Doing so will provide insight into which parameters have a significant effect on the MOEs. With this analysis, the output can then be mapped back to the values of the decision components to derive any possible logical patterns that can show the effects of different agent personalities. These personalities may be able to be related to the factors attributing to unit cohesion.

d. Future Research:

i. **Exploration of More Force Sizes for Emergent Patterns:** The thesis managed to examine only two force size levels for possible emergent patterns. More levels of force size can be examined.

ii. **Effects of Transition Width of the Value Components:** In this thesis, only the priorities in the value components of the movement decision were examined. Future exploration may include looking at the effects of the transition width of each value component, just as the priorities were examined. The exploration can also be extended to include the tactic0/tactic1 decisions.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX I. SCENARIO FILE FOR SMALL FORCE SIZE


```xml
<?xml version="1.0" encoding="UTF-8"?>
<specification>
 <dataFarmingBlock>Data Farming information</dataFarmingBlock>
 <!-- Farmable inputs -->
 <Sensor>
  <name>BlueSensor</name>
  <range>300.0</range>
  <frameTime>1.0</frameTime>
  <timeout>20.0</timeout>
 </Sensor>
 <Sensor>
   <name> BlueCommanderSensor </name>
   <range>300.0 </range>
   <frameTime> 1.0 </frameTime>
   <timeout> 20.0 </timeout>
  </Sensor>
 <Weapon>
  <name>BlueWeapon</name>
  <range>250.0</range>
  <radius>0.2</radius>
  <pK>0.2</pK>
  <frameTime>1.0</frameTime>
 </Weapon>
 <Movement>
  <name>BlueMovement</name>
  <maxSpeed>3.0</maxSpeed>
  <frameTime>0.1</frameTime>
 </Movement>
 <MovementComponents>
  <name>BlueMovementComponents</name>
  <commanderTrust>0.574</commanderTrust>
  <!--  label, importance multiplier, width, endLabel -->
  <formation>0.526 50.0</formation>
  <tgtInWpnRng>0.684 200.0</tgtInWpnRng>
  <tgtInSnsRng>0.968 200.0</tgtInSnsRng>
  <outHostWpnRng>0.163 200.0</outHostWpnRng>
  <cmdrInSnsRng>0.147 200.0</cmdrInSnsRng>
 </MovementComponents>
 <Tactic0Components>
  <name>BlueTactic0Components</name>
  <formation>0.8</formation>
  <!--  label, importance multiplier, width, endLabel -->
  <cmdrInSnsRng>0.8 200.0</cmdrInSnsRng>
  <spacing>0.6 200.0</spacing>
  <hold>0.8 200.0</hold>
  <observe>0.7 500.0</observe>
  <evade>0.3 2</evade>
  <notBeSurrounded>0.4 0.1</notBeSurrounded>
  <attack>0.8 200.0</attack>
  <amass>0.8 0.8</amass>
 </Tactic0Components>
```

```
<Sensor>
  <name>RedSensor</name>
  <range>300.0</range>
  <frameTime>1.0</frameTime>
  <timeout>20.0</timeout>
 </Sensor>
  <Sensor>
    <name> RedCommanderSensor </name>
    <range>300.0 </range>
    <frameTime> 1.0 </frameTime>
    <timeout> 20.0 </timeout>
 </Sensor>
<Weapon>
  <name>RedWeapon</name>
  <range>250.0</range>
  <radius>0.2</radius>
  <pK>0.1</pK>
  <frameTime>1.0</frameTime>
</Weapon>
<Movement>
  <name>RedMovement</name>
  <maxSpeed>3.0</maxSpeed>
  <frameTime>0.1</frameTime>
</Movement>
<Movement>
  <name>RedStop</name>
  <maxSpeed>0.01</maxSpeed>
  <frameTime>1000</frameTime>
</Movement>
<MovementComponents>
  <name>RedMovementComponents</name>
  <commanderTrust>0.5</commanderTrust>
  <!--  label, importance multiplier, width, endLabel -->
  <formation>0.5 50.0</formation>
  <tgtInWpnRng>0.7 200.0</tgtInWpnRng>
  <tgtInSnsRng>0.7 200.0</tgtInSnsRng>
  <outHostWpnRng>0.4 200.0</outHostWpnRng>
  <cmdrInSnsRng>0.5 200.0</cmdrInSnsRng>
</MovementComponents>
<Tactic0Components>
  <name>RedTactic0Components</name>
  <formation>0.5</formation>
  <!--  label, importance multiplier, width, endLabel -->
  <cmdrInSnsRng>0.5 200.0</cmdrInSnsRng>
  <spacing>0.6 200.0</spacing>
  <hold>0.8 200.0</hold>
  <observe>0.5 500.0</observe>
  <evade>0.3 4</evade>
  <notBeSurrounded>0.4 0.1</notBeSurrounded>
  <attack>0.5 200.0</attack>
  <amass>0.5 0.5</amass>
</Tactic0Components>
<!-- Non-farmable Inputs -->
<CommDevice>
  <name>Blue1</name>
```

```
  <range>500.0</range>
  <channels>1</channels>
</CommDevice>
<CommDevice>
  <name>Blue1and2</name>
  <range>1500.0</range>
  <channels>1 2</channels>
</CommDevice>
<CommDevice>
  <name>Blue2</name>
  <range>1500.0</range>
  <channels>2</channels>
</CommDevice>
<CommDevice>
  <name>Red3</name>
  <range>500.0</range>
  <channels>3</channels>
</CommDevice>
<CommDevice>
  <name>Red3and4</name>
  <range>1500.0</range>
  <channels>3 4</channels>
</CommDevice>
<CommDevice>
  <name>Red4</name>
  <range>1500.0</range>
  <channels>4</channels>
</CommDevice>
<agent>
  <name>blueGrunt</name>
  <side>BLUE</side>
  <sensors>BlueSensor</sensors>
  <weapons>BlueWeapon</weapons>
  <commDevices>Blue1</commDevices>
  <movement>BlueMovement</movement>
  <decisions>viewSensors weaponTarget weaponFire movement</decisions>
  <movementComponents>BlueMovementComponents</movementComponents>
  <commInterval>20.0</commInterval>
  <targetValue>1.0</targetValue>
</agent>
<agent>
  <name>redGrunt</name>
  <side>RED</side>
  <sensors>RedSensor</sensors>
  <weapons>RedWeapon</weapons>
  <commDevices>Red3</commDevices>
  <movement>RedMovement</movement>
  <decisions>viewSensors weaponTarget weaponFire movement</decisions>
  <movementComp onents>RedMovementComponents</movementComponents>
  <commInterval>20.0</commInterval>
  <targetValue>1.0</targetValue>
</agent>
<agent>
  <name>blueLeader</name>
  <side>BLUE</side>
```

```xml
  <sensors>BlueSensor</sensors>
  <weapons>BlueWeapon</weapons>
  <commDevices>Blue1and2</commDevices>
  <movement>BlueMovement</movement>
  <decisions>viewSensors weaponTarget weaponFire movement tactic0</decisions>
  <movementComponents>BlueMovementComponents</movementComponents>
  <tactic0Components>BlueTactic0Components</tactic0Components>
  <commInterval>20.0</commInterval>
  <targetValue>5.0</targetValue>
</agent>
<agent>
  <name>redLeader</name>
  <side>RED</side>
  <sensors>RedSensor</sensors>
  <weapons>RedWeapon</weapons>
  <commDevices>Red3and4</commDevices>
  <movement>RedStop</movement>
  <decisions>viewSensors weaponTarget weaponFire movement tactic0</decisions>
  <movementComponents>RedMovementComponents</movementComponents>
  <tactic0Components>RedTactic0Components</tactic0Components>
  <commInterval>20.0</commInterval>
  <targetValue>5.0</targetValue>
</agent>
<agent>
  <name>redLeaderReinf</name>
  <side>RED</side>
  <sensors>RedSensor</sensors>
  <weapons>RedWeapon</weapons>
  <commDevices>Red3and4</commDevices>
  <movement>RedMovement</movement>
  <decisions>viewSensors weaponTarget weaponFire movement tactic0</decisions>
  <movementComponents>RedMovementComponents</movementComponents>
  <tactic0Components>RedTactic0Components</tactic0Components>
  <commInterval>20.0</commInterval>
  <targetValue>5.0</targetValue>
</agent>
<agent>
  <name>blueCommander</name>
  <side>BLUE</side>
  <sensors>BlueCommanderSensor</sensors>
  <weapons>BlueWeapon</weapons>
  <commDevices>Blue2</commDevices>
  <movement>BlueMovement</movement>
  <decisions>viewSensors weaponTarget weaponFire movement tactic1</decisions>
  <movementComponents>BlueMovementComponents</movementComponents>
  <commInterval>20.0</commInterval>
  <targetValue>10.0</targetValue>
</agent>
<agent>
  <name>redCommander</name>
  <side>RED</side>
  <sensors>RedCommanderSensor</sensors>
  <weapons>RedWeapon</weapons>
  <commDevices>Red4</commDevices>
  <movement>RedStop</movement>
```

```
  <decisions>viewSensors weaponTarget weaponFire movement tactic1</decisions>
  <movementComponents>RedMovementComponents</movementComponents>
  <commInterval>20.0</commInterval>
  <targetValue>10.0</targetValue>
</agent>
<!--Engagement specification -->
<scenario>
  <endTime>1200.0</endTime>
  <!--Blue Force -->
  <!--Blue section 1 -->
  <agent>
    <index>1</index>
    <name>blueGrunt</name>
    <position>-800.0 -1000.0 0.0</position>
    <velocity>2.0 0.0 0.0</velocity>
  </agent>
  <agent>
    <index>2</index>
    <name>blueGrunt</name>
    <position>-800.0 -950 0.0</position>
    <velocity>2.0 0.0 0.0</velocity>
  </agent>
  <agent>
    <index>3</index>
    <name>blueGrunt</name>
    <position>-800.0 -900 0.0</position>
    <velocity>2.0 0.0 0.0</velocity>
  </agent>
  <agent>
    <index>4</index>
    <name>blueGrunt</name>
    <position>-800.0 -850 0.0</position>
    <velocity>2.0 0.0 0.0</velocity>
  </agent>
  <!--Blue section 1 Leader -->
  <agent>
    <index>5</index>
    <name>blueLeader</name>
    <position>-900.0 -925 0.0</position>
    <velocity>2.0 0.0 0.0</velocity>
    <subordinates>1 2 3 4</subordinates>
  </agent>
  <!--End of Blue section 1 -->
  <!--Blue section 2 -->
  <agent>
    <index>6</index>
    <name>blueGrunt</name>
    <position>400.0 -75.0 0.0</position>
    <velocity>2.0 0.0 0.0</velocity>
  </agent>
  <agent>
    <index>7</index>
    <name>blueGrunt</name>
    <position>400.0  -25 0.0</position>
    <velocity>2.0 0.0 0.0</velocity>
```

```
</agent>
<agent>
  <index>8</index>
  <name>blueGrunt</name>
  <position>400.0 25 0.0</position>
  <velocity>2.0 0.0 0.0</velocity>
</agent>
<agent>
  <index>9</index>
  <name>blueGrunt</name>
  <position>400.0 75 0.0</position>
  <velocity>2.0 0.0 0.0</velocity>
</agent>
<!--Blue section 2 Leader -->
<agent>
  <index>10</index>
  <name>blueLeader</name>
  <position>300.0 0 0.0</position>
  <velocity>2.0 0.0 0.0</velocity>
  <subordinates>6 7 8 9</subordinates>
</agent>
<!--End of Blue section 3 -->
<!--Blue section 3 -->
<agent>
  <index>11</index>
  <name>blueGrunt</name>
  <position>-800.0 1000.0 0.0</position>
  <velocity>2.0 0.0 0.0</velocity>
</agent>
<agent>
  <index>12</index>
  <name>blueGrunt</name>
  <position>-800.0 950 0.0</position>
  <velocity>2.0 0.0 0.0</velocity>
</agent>
<agent>
  <index>13</index>
  <name>blueGrunt</name>
  <position>-800.0 900 0.0</position>
  <velocity>2.0 0.0 0.0</velocity>
</agent>
<agent>
  <index>14</index>
  <name>blueGrunt</name>
  <position>-800.0 850 0.0</position>
  <velocity>2.0 0.0 0.0</velocity>
</agent>
<!--Blue section 3 Leader -->
<agent>
  <index>15</index>
  <name>blueLeader</name>
  <position>-900.0 925 0.0</position>
  <velocity>2.0 0.0 0.0</velocity>
  <subordinates>11 12 13 14</subordinates>
</agent>
```

```xml
<!--End of Blue section 3 -->
<!--Blue Commander -->
<agent>
 <index>16</index>
  <name>blueCommander</name>
  <position>-1000.0 -925 0.0</position>
  <velocity>2.0 0.0 0.0</velocity>
  <subordinates>5 10 15</subordinates>
  <mission>
    <type>TRAVEL</type>
    <position>-300 -700.0 0.0</position>
    <velocity>2.0 0.0 0.0</velocity>
  </mission>
</agent>
<!--End of Blue Force -->
<!--Red Force -->
<!--Red section 1 -->
<agent>
  <index>17</index>
  <name>redGrunt</name>
  <position>-500.0 1000 0.0</position>
  <velocity>0.0 0.0 0.0</velocity>
</agent>
<agent>
  <index>18</index>
  <name>redGrunt</name>
  <position>-500 950 0.0</position>
  <velocity>0.0 0.0 0.0</velocity>
</agent>
<agent>
  <index>19</index>
  <name>redGrunt</name>
  <position>-500.0 900.0 0.0</position>
  <velocity>0.0 0.0 0.0</velocity>
</agent>
<agent>
  <index>20</index>
  <name>redGrunt</name>
  <position>-500.0 850 0.0</position>
  <velocity>0.0 0.0 0.0</velocity>
</agent>
<agent>
  <index>21</index>
  <name>redGrunt</name>
  <position>-500.0 800 0.0</position>
  <velocity>0.0 0.0 0.0</velocity>
</agent>
<agent>
  <index>22</index>
  <name>redGrunt</name>
  <position>-500.0 750 0.0</position>
  <velocity>0.0 0.0 0.0</velocity>
</agent>
<!--Red section 1 Leader -->
<agent>
```

```
  <index>23</index>
  <name>redLeader</name>
  <position>-400.0 875 0.0</position>
  <velocity>0.0 0.0 0.0</velocity>
  <subordinates>17 18 19 20 21 22</subordinates>
</agent>
<!--End of Red section 1 -->
<!--Red section 2 -->
<agent>
  <index>24</index>
  <name>redGrunt</name>
  <position>700.0 125 0.0</position>
  <velocity>0.0 0.0 0.0</velocity>
</agent>
<agent>
  <index>25</index>
  <name>redGrunt</name>
  <position>700.0 75 0.0</position>
  <velocity>0.0 0.0 0.0</velocity>
</agent>
<agent>
  <index>26</index>
  <name>redGrunt</name>
  <position>700.0 25 0.0</position>
  <velocity>0.0 0.0 0.0</velocity>
</agent>
<agent>
  <index>27</index>
  <name>redGrunt</name>
  <position>700.0 -25 0.0</position>
  <velocity>0.0 0.0 0.0</velocity>
</agent>
<agent>
  <index>28</index>
  <name>redGrunt</name>
  <position>700.0 -75 0.0</position>
  <velocity>0.0 0.0 0.0</velocity>
</agent>
<agent>
  <index>29</index>
  <name>redGrunt</name>
  <position>700.0 -125 0.0</position>
  <velocity>0.0 0.0 0.0</velocity>
</agent>
<!--Red section 2 Leader -->
<agent>
  <index>30</index>
  <name>redLeaderReinf</name>
  <position>800.0 0 0.0</position>
  <velocity>0.0 0.0 0.0</velocity>
  <subordinates>24 25 26 27 28 29</subordinates>
</agent>
<!--End of Red section 2 -->
<!--Red section 3 -->
<agent>
```

<index>31</index>
<name>redGrunt</name>
<position>-500.0 -750.0 0.0</position>
<velocity>0.0 0.0 0.0</velocity>
</agent>
<agent>
<index>32</index>
<name>redGrunt</name>
<position>-500.0 -800.0 0.0</position>
<velocity>0.0 0.0 0.0</velocity>
</agent>
<agent>
<index>33</index>
<name>redGrunt</name>
<position>-500.0 -850.0 0.0</position>
<velocity>0.0 0.0 0.0</velocity>
</agent>
<agent>
<index>34</index>
<name>redGrunt</name>
<position>-500.0 -900.0 0.0</position>
<velocity>0.0 0.0 0.0</velocity>
</agent>
<agent>
<index>35</index>
<name>redGrunt</name>
<position>-500.0 -950.0 0.0</position>
<velocity>0.0 0.0 0.0</velocity>
</agent>
<agent>
<index>36</index>
<name>redGrunt</name>
<position>-500.0 -1000.0 0.0</position>
<velocity>0.0 0.0 0.0</velocity>
</agent>
<!--Red Section 3 Leader -->
<agent>
<index>37</index>
<name>redLeader</name>
<position>-400 -875 0.0</position>
<velocity>0.0 0.0 0.0</velocity>
<subordinates>31 32 33 34 35 36</subordinates>
</agent>
<!--End of Red section 3 -->
<!--Red Commander -->
<agent>
<index>38</index>
<name>redCommander</name>
<position>-300 -875.0 0.0</position>
<velocity>0.0 0.0 0.0</velocity>
<subordinates>23 30 37</subordinates>
<mission>
<type>VECTOR</type>
<position>-300 0.0 0.0</position>
<velocity>0 0.0 0.0</velocity>

</mission>
</agent>
<!--End of Red Force -->
</scenario>
</specification>

LIST OF REFERENCES

[1] Dr. Andy Ilachinski, Center for Naval Analyses, "Irreducible Semi-Autonomous Adaptive Combat (ISAAC): Artificial Life Approach to Land Combat," Military Operations Research, V5 N3 2000.

[2] Dr. Gary Horne, MITRE, "Background Brief in 4^{th} PAIW," Presentation Slides, 2001.

[3] Maj Russell L. Bergeman, USMC, MCCDC, "Considering the Intangibles: Identifying Social Capital in Military Units," Maneuver Warfare Science, 2001.

[4] Dan Croghan, Emergent Information Technologies, Inc, "SOCRATES: A Tool for Agent-Based Distillation," Presentation Slides, 2001.

[5] Dan Croghan, Emergent Information Technologies, Inc., "Draft SOCRATES User Manual," 2001.

[6] Dan Croghan, Emergent Information Technologies, Inc., "Draft SOCRATES Analyst's Manual," 2001.

[7] George E.P Box, Hunter, William G., Hunter, J. Stewart, *Statistics For Experiments: An Introduction to Design, Data Analysis and Model Building*" John Wiley & Sons, Inc., 1991.

[8] McKay, M.D., Conover, W.J., and Beckman, R.J., "A Comparison of Three Methods for Selecting Values of Inputs Variables in the Analysis of Output from a Computer Code," Technometrics, Vol. 21, 1979.

[9] Data Analysis Production Division, MathSoft Inc., *S-Plus 4: Guide to Statistics*, 1997.

[10] Jay L. Devore, *Probability and Statistics for Engineering & the Sciences (Fifth Edition)*, Brooks/Cole, 2000.

[11] Lawrence C. Hamilton, *Regression with Graphics: A Second Course In Applied Statistics*, Wadsworth, Inc., 1992.

[12] Douglas C. Montgomery, *Design and Analysis of Experiments (Third Edition)*, John Wiley & Sons, Inc., 1991.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Professor Tom W. Lucas
   Department of Operations Research, code OR/LT
   Naval Post Graduate School
   Monterey, CA

4. Dr. Alfred Brandstein
   Study and Analysis Division, Marine Corps Combat Development Command
   Quantico, VA

5. Dr. Gary E. Horne
   MITRE Corporation
   Woodbridge, VA

6. Stephen C. Upton
   MITRE Corporation
   Tampa, FL

7. Dan Croghan
   Emergent Information Technologies, Inc.
   Vienna, VA

8. Wong Chen Guan
   Army Operations Analysis Branch
   Singapore Armed Forces
   Blk 303, Gombak Drive
   #03-12

9. Wan Szu Ching
   Army Operations Analysis Branch
   Singapore Armed Forces
   Blk 303, Gombak Drive
   #03-12