Development and Application of a

Rapid Military Model Development Framework

A Undergraduate Thesis
Presented to
The Academic Faculty

by

Nelson Andriano

In Partial Fulfillment
of the Requirements for the Degree
B.S. in Aerospace Engineering with Research Option in the
School of Aerospace Engineering

Georgia Institute of Technology
December 2010

Development and Application of a

Rapid Military Model Development Framework

Approved by:

Dr. Dimitri Mavris, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Santiago Balestrini-Robinson
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: December 13, 2010

Acknowledgments

I would like to thank and acknowledge Dr. Santiago Balestrini-Robinson for his help and guidance with this research and other academic endeavors as well as Dr. Dimitri Mavris and the Aerospace Systems Design Lab for giving me this opportunity. Additionally, I would like to thank the Office of Naval Research. Without ONR’s support, this research would not have been possible.

Table of Contents

List of Figures

List of Tables

List of Equations

List of Symbols and Abbreviations

A .... number of ships on side A

$A_{m \times n}$ .... shots per salvo per weapon

a_1 .... missiles fired per ship from A to B

a_2 .... the number of missiles fired from A and deflected/destroyed per B

a_3 .... number of missiles needed to destroy a single ship

ABM .... Agent Based Modeling

ACL .... Agent Communication Language

B .... number of ships on side A

$B_{m \times n}$ .... shots prevented/missed per salvo per weapon

b_1 .... missiles fired per ship from B to A

b_2 .... the number of missiles fired from B and deflected/destroyed per A

b_3 .... number of missiles needed to destroy a single ship

BDI .... Beliefs, Desires, and Intentions

DOE .... Design of Experiments

FIPA .... Foundations Intelligent Physical Agents

$G_{m \times n}$ .... targeting matrix

$H_{m \times n}$ .... inverse of shots to kill (weapon and target specific)

ISAAC .... Irreducible Semi-Autonomous Adaptive Combat

JWARS .... Joint Warfare System

$K_{m \times n}$ .... normalized damage taken by each asset from each weapon

LEs .... Lanchester Equations

m .... number of defending assets

MANA .... Map Aware Non-uniform Automata

n .... number of attacking weapons

OSD .... Office of the Secretary of Defense

Abstract

Military operations are complex systems composed of the interactions of many smaller discrete systems, or assets: aircraft, watercraft, troops, etc. Historically, the requirements for new assets have been created based on standalone optimization. It is not just necessary to optimize requirements for a single scenario, such as a wartime operation, but instead to optimize the requirements that will benefit the entire military operation as a whole in a number of different scenarios, such as wartime and peace time. To better define future military assets it is necessary sample a large number of scenarios. To capture all of the interactions and develop a complete understanding of the overall system, it is necessary to model both combat and logistics, which have traditionally been modeled and analyzed separately. To characterize military operations and the assets that contribute to them, it is necessary to move beyond the traditional models that use aggregated approximations for combat and stand alone nodal analysis for logistics. A unique need for a framework which captures the complex interaction between combat and logistics while allowing a large number of automated cases and scenarios to run with no human in the loop. The framework this paper discusses was created to facilitate the making of models to analyze and characterize military operations and the effects that future assets will have on entire operations. The framework is agent-based, allowing bottom up definition and the gathering of emergent behavior, and uses a modified Hughes' salvo method for combat, the Foundation for Intelligent Physical Agents messaging structure, and the beliefs, desires, and intentions (BDI) agent model. The modeling of communication and BDI creates myopic agents that are constrained by the information they can obtain, process, and react to. In this paper, the framework is first depicted and then validated by the creation of a model with the purposes of defining the requirements for a future asset, the Transformable Craft. The creation and testing of

the model prove that the requirements for the framework have been met with success. The potential applications of the framework ranges from data-farming military operations models for future asset requirement, characterizing military operations systems, and providing a stepping stone for future agent-based military operations modeling and simulation work.

Introduction

It is important to understand how the systems of military operations that are designed today (e.g., aircraft, ships, tanks, communication structures, etc.) will affect future military capabilities. Capability based planning and acquisition has become an essential part in ensuring that the final product will possess the desired capabilities and the desired effect when deployed. It is essential to explore a large number of alternatives and possibilities in order to understand the effects a future asset will have and the capabilities that the asset will need to produce these effects What is required is a way to allow the experts to quickly explore ideas and understand the impact that different approaches and different systems will have on the performance of the integrated force. The key gaps addressed in this work are flexibility in the analysis and the ability to explain the behaviors of the integrated force.

The future scenarios where these systems of-systems, or military operations composed of the interaction of military assets, will operate are uncertain and, therefore, it is necessary to study and analyze a large number of scenarios to achieve the desired robustness. Traditionally, these analyses have been conducted within the scope of a wargame or similar basic models. Wargaming models address the effects of combat and higher levels of military decision making. These combat models use aggregated force methods that suffer when the forces being analyzed are small and heterogeneous and, at the same time, these methods rely on historical data for their attrition rates. The large number of assumptions made using current methods as well as the need to understand the impact of specific assets and new operational constructs on the overall operation drive the need for a new method of analysis. A reasonably accurate portrayal of a future asset or operational construct's abilities cannot be acquired through historical data and sufficient detail is not available to fully analyze future operational constructs using current methods. In addition, the high level decision making, currently used in wargaming and in many of today's military operation models, requires a human in the loop. When a large number of scenarios are necessary, it is important to reduce human involvement in the simulation process, which is difficult using current methods and techniques. Decision making as a whole, including both the high-level decisions of a general and local-level decisions of a squad leader, needs to be included to fully review the system of systems as it fulfills the military operation objectives. The automation of the decision-making process is key to allowing a large number of scenarios to be addressed and the robustness of a future system, military asset or operational construct, to be ensured. Current methods also focus on pieces of the military operation, not the military operation as a whole. A military operation includes logistics (e.g., troop movement, consumption, supplies, communication, etc.), combat, and decision making. Traditional and current methods lack the simultaneous modeling of logistics, local decision making, and local combat. These pieces are interdependent and tightly interwoven, making separate analyses of each an incomplete assessment. To get a complete analysis of a military operation, it is necessary to analyze all of the pieces together, which current tools and techniques are incapable of.

To satisfy this need, a framework with the capability to support the study of all the elements of a military operation simultaneously, conduct multi-resolution military modeling, and enable automated decision making at multiple levels is necessary. The final objective is to be able to evaluate how local decision making, rules of engagement, supplies and consumption, communication, limited information, individual asset attributes, etc., will affect the military operation's success as whole. This framework allows the modeling and simulation community to examine and understand the interwoven effects of logistics, combat, and decision making on one another and the overall success of military operations. The framework is agent based, allowing models to be built from the bottom-up from readily available information of individual systems' capabilities rather than historical combat data, e.g., attrition rates, etc. Agent based models also allow analysts to obtain insight into emergent behavior, produced by the complex interactions within system, capturing and adequately modeling the logistic modeling needs. The combat method used consists of a modified Hughes' salvo method and a standard projectile combat method. The communication system will follow the Foundation for Intelligent Physical Agents' (FIPA) messaging structure specifications. Each agent will be equipped with beliefs, desires, and intentions (BDI), which will drive its behavior and interaction with its surroundings. The modeling of communication and BDI creates myopic agents which are constrained by the information they can obtain, process, and react to.

Once completed, the model was used to evaluate the possible requirements for a future transportation watercraft, the Transformable Craft, or T-Craft. To demonstrate the capabilities of the framework, a model of a military operation was created and a large number of parameters defining the model, from troop size, location, and capabilities (specifically the T-Craft) were varied. The framework's ability to easily create and analyze the contribution of a single asset towards the complete mission's success was proven. Using the framework allows the evaluation of capabilities of the future transportation craft to be tied to higher mission needs, dependent upon the military operation as a whole, and not only the performance of itself as an individual asset. In essence, the framework provides a mapping between the key performance parameters and measures of performance of the asset to the measures of effectiveness and force-level measures of effectiveness of the aggregated force.

Chapter I

Literature Review

A thorough literature search of traditional and current military modeling techniques and methods was conducted to first gauge the current needs and then identify the critical gaps and the possible solutions to bridge them.

1.1 Traditional Models

The modeling of military operations falls largely into two categories: the modeling of combat and the modeling of logistics. Typically, these two categories are modeled as separate entities. Combat, as traditionally defined, consists of the engagement of troops including offense, defense, and attrition. Military logistics includes supply consumption, supply distribution, troop movement, asset communication, etc. To realize a military operation, it is necessary to not just understand the logistics and combat separately but to analyze and take into account the interactions of the two.

1.1.1 Combat Models

Traditionally, combat models treat forces as aggregated and homogeneous. The first of such models is the set of predator-prey type differential equations known as the Lanchester Equations (LEs). The LEs are subject to a number of restrictive requirements. The requirements for the application of LEs are: homogeneous forces, continuous battle, invariable firing rates, and complete knowledge of the battle field. There are other issues with the approximations made using the LEs beyond the prerequisite requirments. There is no spatial component, no limited information component, and no individual asset component. These issues raised so far are those which stem from the basic assumptions made in the development of the LEs. (Ilachinski 1999) In addition to the issues that arise from the assumption made in creating the LEs, there are problems with the values used. The attrition rates of troops must be approximated from historic data and loosely applied. This produces highly subjective and possibly highly inaccurate or influenced results. (Crooks and Kandel 1992) There have been models created for heterogeneous forces that still rely on basic principles of the LEs and still have the same inherent inadequacies that the homogeneous models have. The LEs represent a first class of combat model, as differential equations based on historical data.

There are a number of additional aggregated models that have been developed over the last half of the century that fall into a second class of aggregated models, the first class being differential equation based. This second class manly relies on a normalized force or a power index. (Dupuy 1985) The second class of combat models, like the first class of models, is based on interpreted historical data. The results from these are binary, either side A or side B succeeds. Nothing other then the coin flip result can be gained from this style of model. The first class at least has the ability to give an estimate on the attrition but this second category of aggregated models provides no such information. Both of these methods are subject to tailoring of the numbers used in creating specific models and are loosely calibrated on past performance. This makes the analysis subjective and calls into question its validity. When historic data is not available, these models' veracity degrades even further. For that reason, the assessment of future military forces and assets the applicability becomes doubtful. Both of these classes of models can be viewed as top-down approximations with a limited ability to capture the details that characterize combat.

On the opposite end of the spectrum, salvo models provide a radically different approach. Instead of approximating the overall attrition rate as the first two classes of models do, this third class counts each shot fired by each asset and evaluates the effects of each shot on its target. The Hughes' salvo method was originally developed for ship-to-ship combat, in particular heterogeneous torpedo exchanges. (Tiah 2007) It is also possible to capture the uncertainty in both the defense and the attack parameters by representing them as probability distributions. This can be used to model the accuracy, reliability, etc., of individual assets or weapons. (Armstrong 2005) The salvo methods explicitly capture the phenomena of combat in a more detailed level than the use of differential equations or summed power indices. As a drawback, they are more difficult to apply to large scale models both in terms of the amount of modeling effort required and the time required to compute their simulations. An important distinction for the salvo method with respect to the other two, is that it requires spatial awareness and individual asset representation.

1.1.2 Logistics Models

There are two primary methods used to model logistics. The first is the use of graph theory and nodal analysis. In this first method, locations and points of interest, whether they are depots, forward operating bases, or locations of troops, are all portrayed as nodes. The transfer of goods is done along weight edges, where weights are a function of the geography, transportation type, goods being transferred, etc. (Gue 2003) These models approximate spatial, asset, and supply characteristics of the force, making them more analogous to the LE and power index combat models. Traditional questions addressed by these type of models is what is a good routing plan for the logistics forces, how much throughput can be produced by the network, and where are the bottlenecks going to occur.

The second method most commonly used is discrete event simulation. This method consists of modeling a scenario as a process, composed of a set of ordered and sequential events.

Each step in the process takes some quantity of time and demands certain resources, such as a transport ship travelling to shore which consumes gas depending upon its payload, and is preceded and followed by similar discrete events. (Beisecker 2008) The associated time and resources consumed allow the logistics flow to be analyzed. This second method is based directly on asset attributes, fuel consumption, speed, payload limits, etc. The demand, or logistic load, depends on the current combat situation which must be assumed during both the discrete event simulation and the graph theory analysis. The inherent interdependency between logistics and combat is not explicitly accounted for in either one of these methods. Some discrete event simulations will attempt to capture the combat contribution by the logistics force, but this is generally done using a combat power index, which is an abstraction of the combat power, and one that tends not to be suitable to capture the complexities of combat.

1.2 Current Modeling Needs

The reasons for modeling and simulating warfare are well summed up in the goal statement behind the creation of the Joint Warfare System (JWARS). Its goal was to provide "a simulation of joint warfare that will support operational planning and execution, force assessment studies, system trade analyses, and concept and doctrine development." JWARS was a joint military program to create a military operations simulation and modeling framework. The need to predict an outcome to help plan and execute a specific mission, the need to assess the capabilities of a given force and the need to develop future doctrine are the central reasons to develop and utilize such a tool. The fourth reason, system trade analysis, has become more critical in recent decades with the increased reliance on system acquisition analyses based on constructive simulations. Examining the force conducting a given scenario as a system of systems, enables not only the planning and execute an operation, assessing the capabilities of a force, or defining doctrine but more importantly in planning how the future force will be employed. The iteration of what means to acquire and the ways to employ those means makes capability-based acquisition an intractable problem. A flexible simulation tool can ease the different excursions and allow experts to test more plans and make more informed decisions.

Historically, the Department of Defense (DoD) has relied on a mixture of qualitative expert judgment and quantitative analysis to identify the required capabilities for future systems. This approach has proved less effective as the complexity of the environments in which the future asset will be used increases. As the complexity increases, humans' ability to predict behavior decreases and it becomes more important to be able to examine the future assets' impact on the differing environments in which it will be placed. Only by examining the effects a future asset will have on the environments in which it will be inserted can the most robust set of requirements be defined. In other words, the required capabilities of a future asset should be driven by the needs generated from operating in all possible environments and with any combination of systems with which it may operate. The level of analysis necessary to define the capabilities of a future asset far surpasses those provided by qualitative expert opinion due to the complexity of the problem. To effectively analyze the asset's impact at the system-of-systems level, the mapping between the asset's performance and the overall behavior of the SoS must be quantified, current methods do not excel at providing this capability. In addition to examining the performance of differing systems containing the future asset, it is important to examine the sensitivity the performance of each system to the capabilities of the future asset. Agent based modeling has been employed to help capture this impact and there exist a number of ways to help quantify it.

1.3 Agent Based Modeling

Agent based modeling works much like the salvo combat methods and the discrete event simulation in that it is built up from individual asset attributes and characteristics to produce an encompassing model. Agent based models (ABMs) are created on the idea that a system is best represented as the integration of its sub-systems and components. The rules governing ABMs are the rules that govern the behavior of individual agents. The interaction between agents and their surroundings produces the macro-level behaviors. In contrast, traditional modeling is done by using the macroscopic behavior directly based on microscopic attributes and macroscopic trends. (Ilachinski 1999)

Ilachinski (Ilachinski 1999) was one of the first researchers to employ ABM to model combat. His first models, ISAAC (Irreducible Semi-Autonomous Adaptive Combat) and EINSTien (Enhanced ISAAC Neural Simulation Toolkit), studied how simple meta-rules can produce some of the complex patterns observed in actual combat situations. His work demonstrated that simple meta-rules for how each individual asset would react to their current situation, including current location, surroundings, objective, and well being resulted in complex emergent behaviors. In essence, a personality was created for each of the agents present in the simulation. The rules governing the model produced expected behavior from the agents involved in the simulation including surrounding enemies, punching through front lines, and flanking. (Ilachinski 1999) T ISAAC and EINSTien were later evolved into a more comprehensive combat simulation tool called MANA (Map Aware Non-uniform Automata). MANA added functionality into the behaviors of the agents, but it has been found that it is difficult to tailor its capabilities to scenarios and missions that diverge from its original intent, e.g., MANA does not address current needs such as large scale logistics and complete C4ISR capabilities, an analysis of a large number of scenarios with ease, and a prediction of future asset performance. MANA works well when used for well defined situations but it becomes cumbersome to use it for scenarios that require more complex decision making processes from the agents or higher complexity behaviors. MANA is not a flexible framework and can be very hard to modify to specific needs. There have been a number of other frameworks produced for military modeling, each having its strengths and weaknesses.

The next simulation framework to discuss is the Joint Warfare System (JWARS) has been in development and some use for near a decade. The program was funded by Office of the Secretary of Defense (OSD) to be used by the OSD, the Joint Staff, the Services, and the Combat(ant) Commands. JWARS was one of the first models attempted to integrate all of the elements of a military campaign from planning to execution. The simulation can be run from doctrines, rules of engagement, and campaigns while incorporating troop location and movement, logistics and geography. This agent based model represents a leap from the aggregated troop attrition models previously discussed but that does not signify that there are no drawbacks. Though JWARS allows a look into the entire combat system, the use of the simulation framework is restricted and not excessively customizable by the user. Furthermore as far as can be elucidated from the open literature, the final version of JWARS was not released as of yet.

Private companies have developed unrestricted simulation frameworks such as FLAMES. This product, by the Ternion Corporation, is available for purchase as a military combat modeling and simulation framework. It has the flexibility to be modified to cater to a given need. The flexibility of the framework does not take away from FLAMES' ability to incorporate details where other simulation methods would require assumptions. FLAMES may seem like the optimum choice for a non-military institution to use, but there are drawbacks in the cost and the development time. While the program is not restricted by security concerns, it is restricted by the cost of a license to use. In addition, FLAMES allows for the parameterization of given scenarios once they are created, but the lead time for creating the individual scenarios is considerable. The information that is used for agents is only available in restricted databases.

1.4 Summary

The literature search of current capabilities for military modeling and simulation has shown a distinct need for a framework that can be easily modified to fit specific needs and can facilitate the rapid development of combat models to allow for the analysis of a military operation as a whole, instead of a sum of pieces as traditionally done. This framework needs to be unique in that it can be used to quickly create a combat scenario or that it can be modified to fit any differing need a user may have while keeping a shallow learning curve. The current deficiencies of military operation modeling and how each deficiency will be addressed with the development of the new framework are summarized in Table 1.

Table 1. Deficiencies of Current Capabilities and Proposed Solution Methods

Deficiencies of Current Capabilities	Proposed Solution Methods
Non-aggregated combat	Hughes' salvo method
Microscopic information driven	Agent based modeling
Incorporate combat and logistics	Agent based modeling and Hughes' salvo method
Short scenario development time	NetLogo
Parameterization of scenario initial condition	NetLogo
Open source and easily modified framework	NetLogo
Automated decision making	General agent decision engine

In summary, agent based models are built based on the microscopic information of individual agents and allow the analyst to observe some of the plausible macroscopic behaviors that the integrated force may exhibit. The Hughes' salvo method provides a more realistic approach to combat than the simplified aggregated models. The agent based modeling techniques will also allow the consumption and distribution of supplies to be fully realized while the Hughes' salvo method will do the same for the combat, allowing combat and logistics to be studied side by side. Creating the open source framework in NetLogo will allow for the creation of multiple scenarios, the parameterization of said scenarios, while allowing for the simple modification of the framework itself. A general agent decision engine will also be created to automate the local and high level military decisions.

Chapter II

Methods and Tools

2.1 Agent Based Modeling and NetLogo

Agent based modeling works by building up from individual asset attributes and characteristics to produce an encompassing model. Agent based models (ABMs) are created on the idea that a system is best represented as the integration of its sub-systems and components. The building block of ABM is the agent, in essence, an entity that changes its state and possibly the state of its surroundings based on a subset of the states of the agents and the environment, i.e. based on the subset of the environment that the acting agent perceives. In general, agents are myopic, meaning that they can only observe their immediate surroundings, and oftentimes, agents are modeled as manipulating imperfect information. They may observe information incorrectly, or they may receive information from other agents incorrectly. They change their states based on internal rules which may be as simple as if-then statements, or as complex as adaptive neural networks. The rules governing ABMs are the rules that govern the behavior of individual agents, that is not to say that a higher level agent may not mandate an action to a low level agent, but from the implementation standpoint, that mandate can be interpreted as an observable state to the lower ranking agent, who in turn changes its state to the state mandated by its superior. The interaction between agents and their surroundings produces the macroscopic behavior that is of interest to an analyst interested in capture the emergent behavior of the system. In contrast, traditionally, modeling is done by modeling the macroscopic behavior directly based on microscopic attributes and macroscopic trends, e.g., system dynamics, process modeling, etc. (Ilachinski 1999)

The agent based modeling platform on which to build the framework was a key decision. The main requirements for the modeling tool were for it to be a software product with a repeatable trajectory, to have current and available support, to be free to download and use, and to be easy to learn. Table 2 contains the assessment of the four frameworks evaluated, RePast, MASON, NetLogo and FLAMES. NetLogo was chosen because it was the only framework that satisfied all the requirements. NetLogo is a free to use multi-agent programmable modeling environment. It is under current continuous development by The Center for Connected Learning and Computer-Based Modeling. It is available to any person who may want to use it, along with a large number of tutorials and references. It is under current development, meaning that there is ongoing improvements and support for the modeling environment. The language used is a Logo-based language, which was originally developed to be easy to learn by being intuitive in its lexicon and semantics. NetLogo itself is built on Java and the developed models can be compiled and shared without the need of the NetLogo program. Furthermore, a vibrant user community exists around NetLogo, and new extensions and models are being constantly developed and shared freely. NetLogo is also easily combined with MATLAB, Mathematica, and ModelCenter to aid in its automation. For these reasons, NetLogo was the ideal candidate for the platform for our rapid military operations model framework.

Table 2. Agent-based Modeling Frameworks

Criteria	RePast	MASON	NetLogo	FLAMES
Trajectory	**Excellent**	Good	Very Good	Good
Support	Good	Discontinued	**Very Good**	Good
Cost	**Free**	**Free**	**Free**	Expensive
Learning Curve	Very Steep	Steep	**Very Shallow**	Moderately Steep

2.3 Beliefs, Desires, and Intentions

The key to an agent based model is defining and modeling the actions of the individual agents, because from the individual agents the emergent behavior of the entire system is obtained. There are multiple ways to model an agent's behavior, from an ad hoc method, to more rigorous methodologies that help the developers structure the agents and their methods, helping produce more rigorous and effective agents. The Beliefs, Desires, and Intentions (BDI) method was chosen to model the individual agents in this case because it provides an easy to understand set of abstractions that not only organize the programming of the agents' behaviors, but is also expandable to reproduce complex human-like decision processes (Georgeff, Pell et al. 1999). In the BDI model, every agent possesses beliefs, desires, and intentions (Figure 1). Beliefs hold the information that the agent has either collected itself through sensing or received through communication from another agent. The beliefs represent all of the facts that go into making decisions, and they may be accurate or inaccurate, allowing the analyst to study the impact of imperfect information on the behavior of the agent. The desires of an agent represent what the agent wants to have happen or wants to do. The desires of an agent can be the result of a direct command, self preservation, default action, etc., using the information it has and its wants/needs contained in the beliefs and desires, the agent them makes a decision on a course of action. The course of action then needs to be carried out as a set of sequential intentions the agent follows. Intentions are simple tasks that can be carried out by the agent, and must he formulated as a method the agent must follow, e.g., travel to coordinate (X,Y), and a condition that indicates when to stop executing said intention and move on to the next, e.g., distance to (X,Y) = 0. The BDI model is well suited for agents that sense their environment, have a set of decision options to make based on the information collected, and then act based on combinations of simple actions. The agents in turn affect their environment and the other agents by acting their intentions.

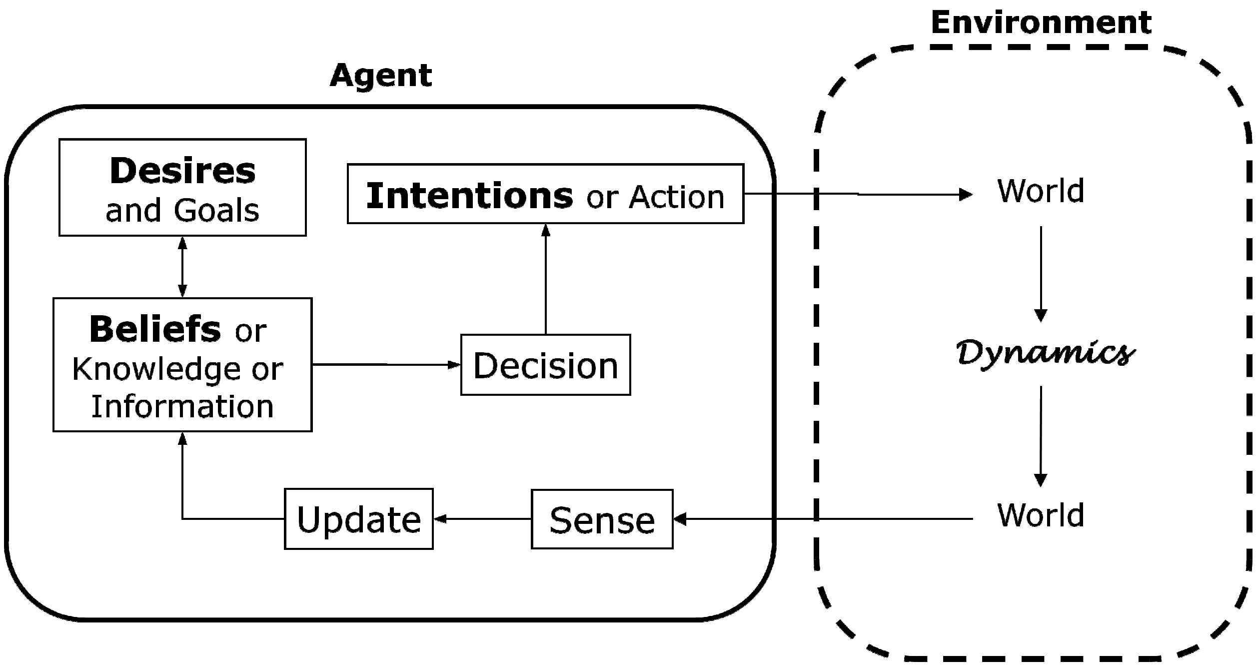

Figure 1. Belief, Desire, Intention Diagram

Three systems were created to help manage the BDI attributes of the agents, one for beliefs, one for intentions, and one, indirectly, for desires. Each belief is stored with a "belief-type" and a "belief-content". Beliefs can be added, read, and removed by the agent depending on the current situation. All sensing and communication is set up to provide beliefs, or information, for the agent to make informed decisions, or uninformed when the provided information is not ideal or imperfect. The desires currently filter through the commands and self preservation wants of the agent, they are not directly responsible for actions. The command structure is discussed in greater detail further on, but at this point, it will suffice to say that it uses the beliefs the agent has to create a plan of action. This action could be to move to another point, to engage an enemy, to not engage an enemy, etc.

To implement the intentions, each agent is given an intention queue (intentions). The intentions are not received externally, but are created based on the beliefs and desires of the agent and what it plans on carrying out. Individual intentions consist of an intention-name and an intention-done. The intention-name links to a repetitive action that is carried out over and over until the intention-done is triggered to be true. Once the intention is completed as stated, the current intention is removed and the next intention executed. There are a limited number of intentions but more can easily be added and incorporated in a given model.

Intentions can be stand alone repetitive tasks, but more commonly they are created from a set of basic abilities. These basic abilities are generic to all agents of all types. Different agents can use the same basic abilities. Each agent creates their own intentions, or courses of action, from their available basic abilities. These courses of action are obtained from decisions made based on the agents' beliefs, or information, and its desires, or wants and needs.

2.4 Defining Agents

A large part of agent based modeling depends on how the assets are defined, or how different asset types abilities and qualities are quantified and included in the model. Figure 2 shows some of the basic information that defines the different types of assets in the framework.

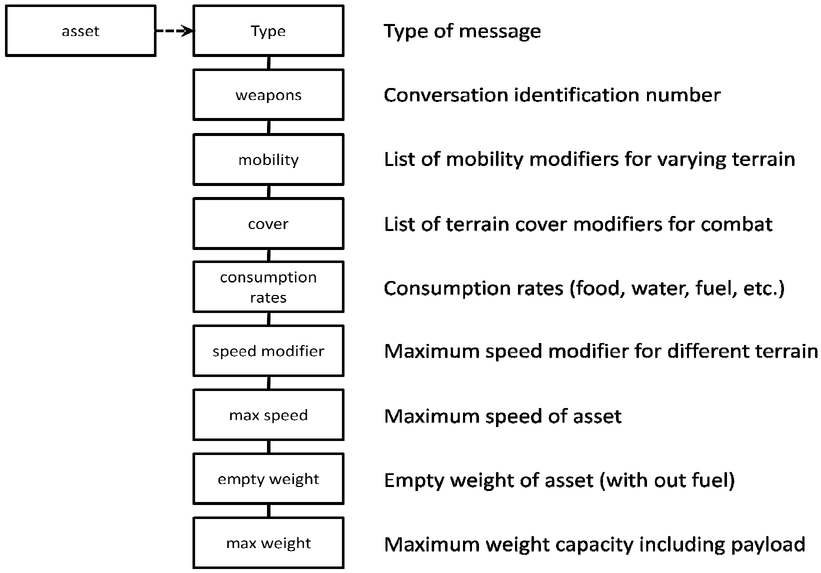

Figure 2. Generic Asset Definition

The asset-type is the string identifier for the given type of asset. The "weapons" is a list of weapons that are at the disposal of the asset during combat. The "mobility" and "cover-list" are modifiers that are used to determine the defense matrix while conducting combat calculations. The "consumption-rates" of an asset depict the consumption of food, water, and fuel. Fuel is consumed per distance travelled while food and water are defined on an amount per day. The daily food and water intake is then consumed at specified intervals of time. Ammunition is not consumed by a rate, but by the usage of the weapons the asset has. The "max-speed" of an asset is specified and then modified using the "speed-modifier" which is defined for different terrain types. For logistics reasons, the maximum-weight and the empty-weight for each asset are defined as well.

2.5 Messaging System

In 2002, the Foundation for Intelligent Physical Agents (FIPA) released an Agent Communication Language (ACL) messaging structure (Agents 2002). The protocols were created to allow efficient and complete communication between intelligent agents. The FIPA

ACL message structure is defined by a number of parameters for a given message. The parameters that were needed to define a message with the framework were largely dependent upon the ACL message standard. depicts the agent messaging structure used in the framework.

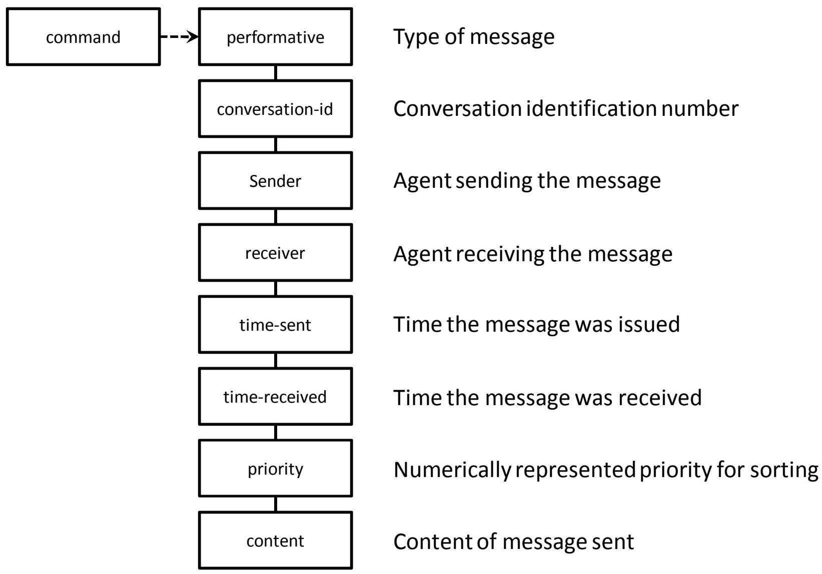

Figure 3. Agent Messaging Structure

Messages sent between agents are defined by the parameters listed in Figure 3. Every agent possesses an incoming queue of messages. When a message is sent, the sending agent asks each of the receivers to receive the message. Receiving the message adds it to the end of the incoming queue of messages of the receiving agent. Sending a message records the message as sent. Each message has a communication delay, time-stamped into the message as a time it can be read (time-received). When the current time equals or exceeds this time stamp, the agent can read the message. When reading the message, depending on what the performative is, the message is processed. When the message cannot be processed due to an unknown performative, a “not-understood” message is sent in reply.

The message system has been incorporated in the belief, desires, and intentions model chosen to represent the agents. The message style to be used largely in the framework is the use of the “belief” performative. This is the only message performative; any other performatives that may be desired by a user can be easily incorporated in the processing function. When a “belief” message is received, the content of the message is automatically included in the beliefs of the agent. The beliefs that are received are then used by other functions, structures, and systems within a given framework and/or model. As an example, commands are sent in the forms of beliefs and then read into an assets command queue and enemy information can be acquired as a belief and stored read and stored with all other enemy intelligence.

The goal of the message system is to create imperfect agents. If all communication between agents is done through the message structure and there is no direct link, the only information the agent receives is from its own sensing techniques and the messages from other agents. Further communication modification can be made. A possible extension could be to include types of communication (direct, radio, satellite etc.) and the vulnerabilities of those communications during combat. The user can use the baseline message structure to expand it far beyond sending and receiving with delays. The benefit of having compartmentalized modules such as the FIPA-ACL communications module, is that other developers have to expend the majority of their efforts improving that module, without requiring them to modify the entire framework.

2.6 Command System

Commands are the key to implementing a hierarchical organized fighting force. The commands are sent via the messaging system as all communication between agents is conducted. The basis of the command structure is the command itself, represented in Figure 4.

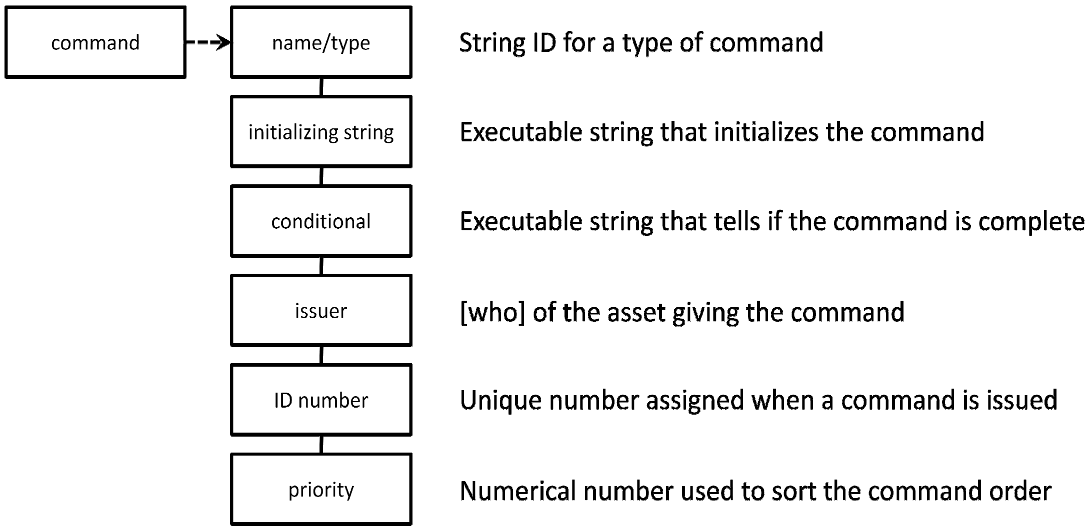

Figure 4. Command Structure Diagram

The command's type and ID number are identifiers. The initializing string is run when the command is first read to set up the intentions and information needed to carry out the command. The conditional is run each tick before carrying out the command at each time step. The issuer is saved for response purposes when the command is completed and the priority is given by the sender to rate the relative importance of commands.

The command process in which they are initialized, carried out, and completed can be seen in Figure 5 and Figure 6.

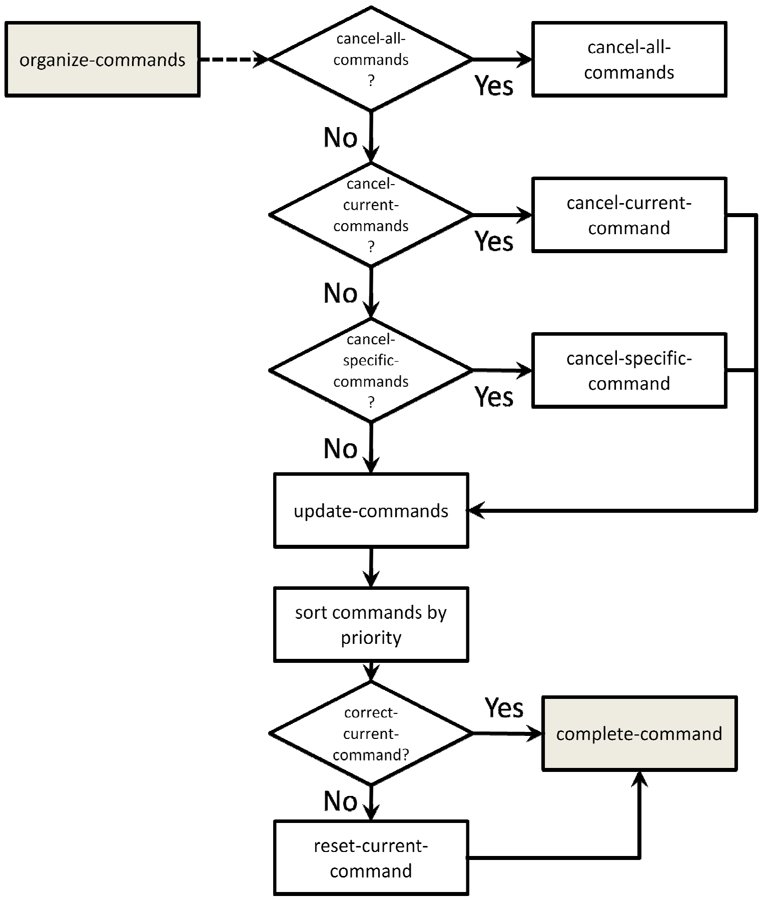

Figure 5. Command Process Flow Chart

Each time step the asset organizes its commands. The first step is to decide whether any of the commands currently in queue need to be removed. This could originate from a message from the assets leader or from completion of the command. There are three options for removal. The asset can remove the current command it is executing, the first in the queue, all of its commands from its queue, or specific commands from its queue. Once this is completed, if there are commands left, the asset will update its commands. When updating its command queue the asset reads in new commands, after which it will sort the commands by priority. Once sorted, if the current command, the command from the previous time step, is still the priority command it will carry it out otherwise it will load the new first in queue command. The asset then begins to carry out the command, or complete the command.

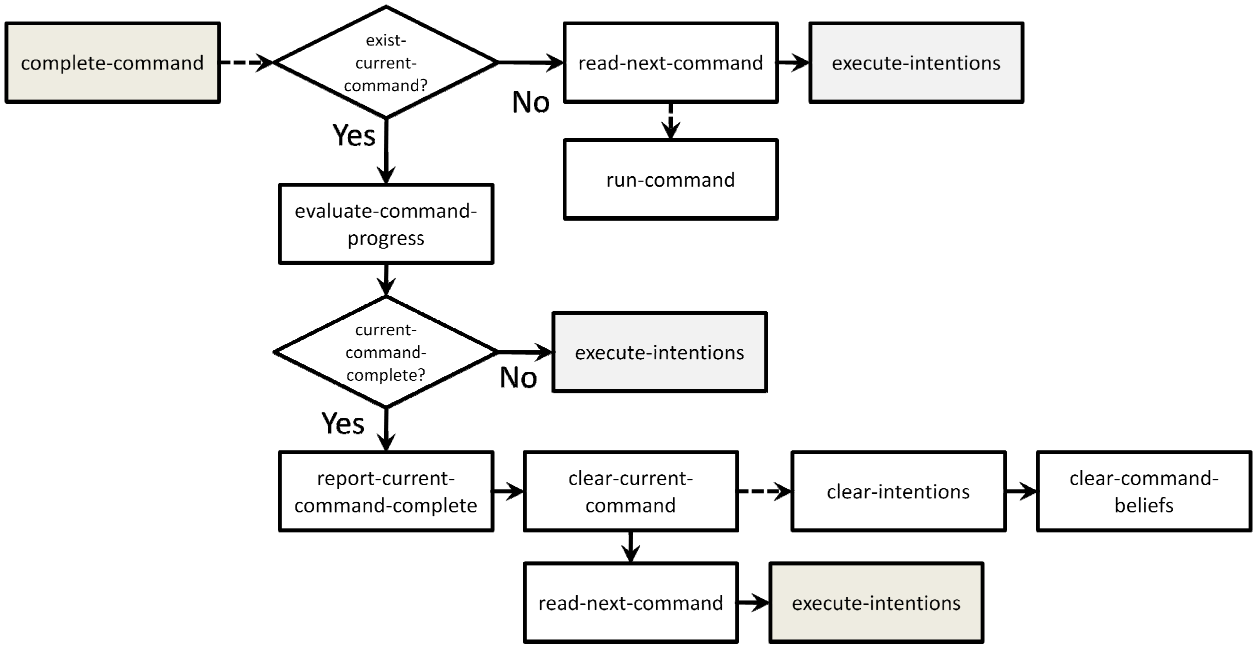

Figure 6. Command Completion Flow Chart

The asset first check to make sure there is a command to complete and it is loaded, if not it loads the next command and runs it to load the new command and then executes the intentions. If the command has already been run before, it first checks to see if the command is complete and if it is not, executes the intentions that the command is made of. If the command is complete, it first clears all of the temporary variables it stores for the command then reads the next command and executes the newly produced intentions.

2.7 Logistics Modeling

An important part of modeling military operations is modeling the logistics of operations. The logistics process begins with the consumption of goods. Thus far, the supplies taken into account include food, water, fuel, and ammunition. Food and water are consumed at specified times of day in an amount based on asset attributes. Fuel is consumed as assets travel and ammunition is consumed as assets engage in combat. As supplies are consumed, the assets will consume down to a threshold point and they will request for supplies. Generic cargo container agents are created and passed between cargo carriers until they reach the specified asset with the request. A cargo container agent is simply a list of supplies (type and quantity) and a total size and weight. They can be made to have the same footprint of a TEU, pallet, etc. The logic for the logistics model is simple and leverages the same methods and techniques described previously. Agents have beliefs which help them decide when to request more cargo. The threshold can be made to vary based on the belief of the agent (e.g., an agent that is further away from a FOB will request cargo at a higher threshold than an agent that is closer by). At the same time, the logistics planner agent issues commands to cargo carriers (e.g., HMMWVs, Mk23s, Mk48s, etc.) and they execute their mission as a set of intentions, e.g., travel to coordinate pair (X_1,Y_1), deliver cargo (Z_1), travel to coordinate pair (X_2,Y_2), deliver cargo Z_2, etc.

2.8 Combat System

As explained previously, the combat system created for the model is based on a modification of the Hughes' Salvo model, originally used for ship-to-ship missile combat. The original model used the following model for representing combat between ships with homogeneous missile fire (Tiah 2007):

$$dA = \frac{b_1 * B - a_2 * A}{a_3}$$

$$dB = \frac{a_1 * A - b_2 * B}{b_3}$$

Equation 1. Hughes' Salvo Method

The Hugues' salvo method was used as a basis for developing a new modified salvo method, represented in Equation 2 describes the method:

$$K_{m\times n} = A_{m\times n} * G_{m\times n} * (1 - B_{m\times n}) * H_{m\times n}$$

Equation 2. Modified Salvo Method

The method developed is not aggregated, so each agent involved in the combat as both an attacker and/or a defender must be identified and included. In actuality, it is not the attackers that are identified but, because attackers can have multiple weapons and can attack with multiple weapons at each time interval, the weapons of the attackers are identified. Each weapon will have a percentage of its volley directed at its desired targets. The number of fires per time step, consisting of the firing rate for each weapon constant across a given column, is multiplied by the targeting matrix, G, which consists of the percentage of the salvo targeted at individual agents. Each column in the G matrix adds to 1 and allows the asset to divide its salvo amongst a number of enemies. The resulting *matrix dot multiplication* (a one-to-one multiplication of matrix components, similar to the vector dot product) gives the number of shots fired from the attacking weapons at each defending asset. The matrix B represents the number of shots missed or prevented during the salvo. The values included are specific to a weapon being fired at a given asset and may include or be affected by the attacker or defenders position, the capabilities of the attacker, the armor or defense of the defender, probability of a hit based on firing distance, etc. Dotting the salvos matrix with *(1-B)* accounts for the number of missed or deflected shots. The final matrix, H, is the inverse of the number of shots from a given weapon it will take to kill or incapacitate a given asset. Multiplying by H normalizes the damage done by individual weapons so they can then be added together and K is finally computed (Figure 7).

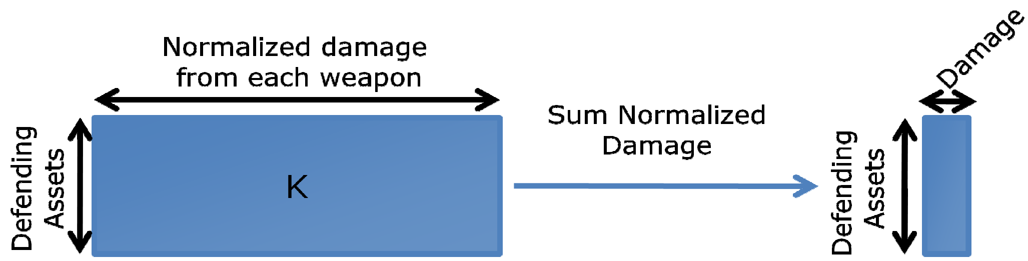

Figure 7. Combat Matrices

Once K is computed, the rows are summed to calculate the total resulting damage received by each defending asset. The damage is subtracted from the assets' current health, which is initialized to a value of 1 (i.e., 100%), and when it reaches zero the asset is killed or destroyed.

The defining value for the weapons used in the combat evaluation is shown in Figure 8 below.

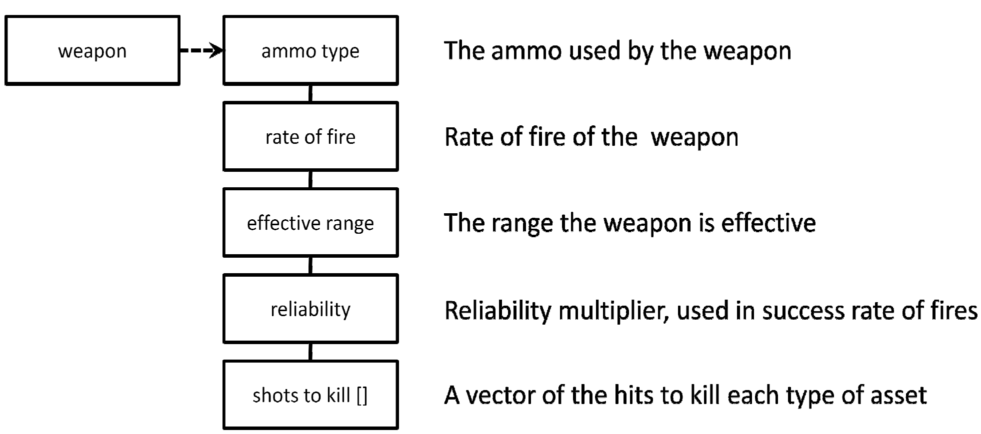

Figure 8. Weapon Definition

The name acts as an identifier. The ammo type is included for logistics purposes and to calculate and track its consumption. The rate of fire is used in constructing the A matrix. The effective range is used in both targeting and calculating the probability of a hit. The reliability is also taken into account when the probability of a miss fire or down time is calculated. The "shots-to-kill" is a vector that describes the number of shots it will take to kill corresponding asset types. (Tiah 2007)

Chapter III

Results

To test the capabilities of the framework, a model was created and used for the analysis of a next generation navy transport concept, the Transformable Craft or T-Craft. Traditionally, new system requirements are defined by addressing known gaps in the current methods employed to conduct operations using the existing accompanying systems and with specific scenarios in mind. The model developed in this case allows a variety of different T-Crafts with specified capabilities to be evaluated while the rest of the scenario is kept constant. This evaluation allows the capabilities of the T-Craft to be measured by studying the behavior of the overall system, and the end state of the entire military operation. For this example, a single-landing combat scenario is examined but it is shown how inserting a single type of T-Craft, or specifying the capabilities of the T-Craft, for both peace and wartime scenarios can allow for a more robust set of requirements to be determined. The framework will allow the model to be evaluated on a single case basis (allowing users to study the detailed behavior of the model) or to be automated, effectively enabling a large number of cases be evaluated and ultimately allowing a large range of combinations of T-Craft capabilities to be compared simultaneously.

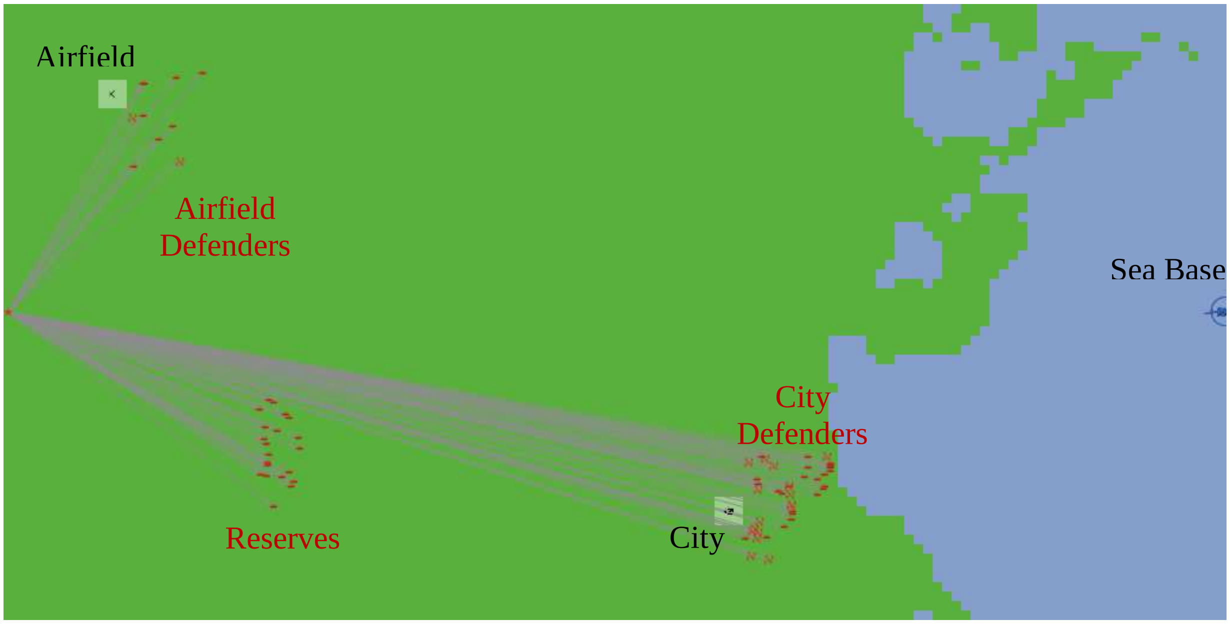

Figure 9. Initial Setup

The model will not evaluate all possible scenarios since such a venture would be unfeasible for the scope if this thesis, but it is designed to characterize the more immediate military force provided by the T-Craft bring forces from the sea during a military landing operation. The model consists of two forces, the defending force, in red, and the attacking force in blue. The goal of the red army is to defend the two military objectives that are present on the shore, a city near the shore and an airfield inland. The blue forces are initially located at a sea base, in essence a large massing of troops and supplies aboard a conglomerate of ships offshore. The T-Craft are used to transport the forces and supplies to shore. The red force consists of three separate groups: those initially set to defend the city and shore, a smaller force set to defend the airfield, and a reserve force stationed inland to be deployed as needed. (Figure 9)

Only a few types of units were selected for the model as it is a demonstrator for the framework. The military forces of each side consist of squad of soldiers and tanks. UAVs are used for reconnaissance. Logistical support is provided through the T-Craft mentioned above for shipping assets and supplies from the sea base to land where shipping trucks are used for further distribution of goods. There are also both red and blue commands present in the simulation that are responsible for the decision-making and commands given during each side's operation.

The red and blue commanders both consist of relatively simple command structures that can be expanded and made more complicated as the modeler's needs. For the demonstration of the framework, the capabilities and complexities of the commander and his decisions were kept to a minimum. Both the red and blue commanders want to gain or maintain control of the military objectives present on the map, namely the city and the airfield. They target the closest or most significant obtainable objective that is not theirs while, with their limited knowledge, they defend the objectives they have already have control over or have captured. The individual assets also have some small decision-making capabilities. They can determine if an attack may be successful depending on the information they have and will retreat to friendly forces around them if the situation does not look favorable and will execute their orders if the situation is favorable. If supplies are needed, assets will suspend aggressive activity and wait for supplies to be delivered and then continue with the current command they have received.

Before automating runs to characterize the entire capability space of the T-Craft, it is necessary to understand the behaviors produced by the model and ensure that the model is producing sensible results. For this purpose, a few results obtained from altering the T-Craft's capabilities will be discussed. To help give a feel for the model created and the cases to be executed there are two scenario walkthroughs below. The difference between the two scenarios is the speed which the T-Craft can travel in open water and the number of T-Craft deployed. This will demonstrate the varying behavior of the system as a whole based on the capabilities of the T-Craft.

3.1 Higher Speed and More T-Craft

Figure 10. UAV Scouting

The red forces are initially just set to defend each of the objectives while the blue forces must first gather intelligence before attacking and landing troops. The blue commander first sends out a set of UAVs to survey different sectors to gather the position of the enemy. (Figure 10)

Figure 11. Blue Forces Land

Once the commander has the needed information, he picks a landing location and begins to send troops to the shore via the T-Craft. (Figure 11)

Figure 12. Red Defends Beach

The red forces react to the landing of the blue forces, sending some assets to protect the shore and slow the assault while others retreat to protect the objective. The forces protecting the airfield also mobilize to defend and protect the objective. (Figure 12)

Figure 13. Blue Attacks Objective

The blue forces overtake the red forces that were sent to protect the beach and move towards the city to take the objective. When the red commander receives the information, it will call for reinforcements to be sent to the city to help. (Figure 13)

Figure 14. Blue Takes Objective and Red Begins Counter

Before the reinforcements arrive, the city is taken by the blue forces. The blue commander does not know the red reinforcements are close to arriving at the city and turns his attention towards the second objective, the airfield. (Figure 14)

Figure 15. Red Reserve Forces Reach Objective

As the red reinforcements reach the city, the blue forces have begun to move toward the second objective, the airfield, more inland. Due to the previous combat, there is a logistical need in the city and the trucks move to deliver the supplies from their drop point on the shore. (Figure 15)

Figure 16. Red Forces Retake the Objective

The blue troops which were left to guard the city are overcome and the red forces counter attack is successful. (Figure 16)

Figure 17. Blue Forces Move to Retake the Objective

When the blue commander learns of this, the blue forces are redirected toward the first objective to retake the city. (Figure 17)

Figure 18. Blue Attacks Red Reserve Forces

The blue forces then attack the city for a second time while putting the assault on the airfield on hold. (Figure 18)

Figure 19. Blue Retakes the Objective

Once the blue forces recapture the first objective, they once again move to take the second objective. (Figure 19)

3.2 Slower Speed and Less T-Craft

Figure 20. Blue Forces Move Toward Objective

The same size force is placed on the beach over a greater amount of time due to the difference in speed and number of T-Craft. (Figure 20)

Figure 21. First Encounter

The first attack is attempted once enough troops have amassed on the beach. (Figure 21)

Figure 22. Red Forces Defend the Objective

Due to the increased waiting time, the red forces reinforcements have already reached the city. (Figure 22)

Figure 23. Red Forces Successfully Defend the Objective

Unlike the previous case, here, the red forces never lose the city during the first attack wave. The blue forces then wait longer to build up for a counter attack unlike before, where there was no need. (Figure 23)

3.3 Initial Comparison of Results

These two different scenarios are presented to give an idea of how a set of capabilities of the T-Craft translate into a change in what happens during the military operation. A crucial metric is the number of blue forces on the beach as a function of time. To measure the difference between the scenarios, a graph is created for each case that shows the percentage of blue forces on the shore as well as the remaining percentage of both forces. The two separate scenario graphs are displayed below.

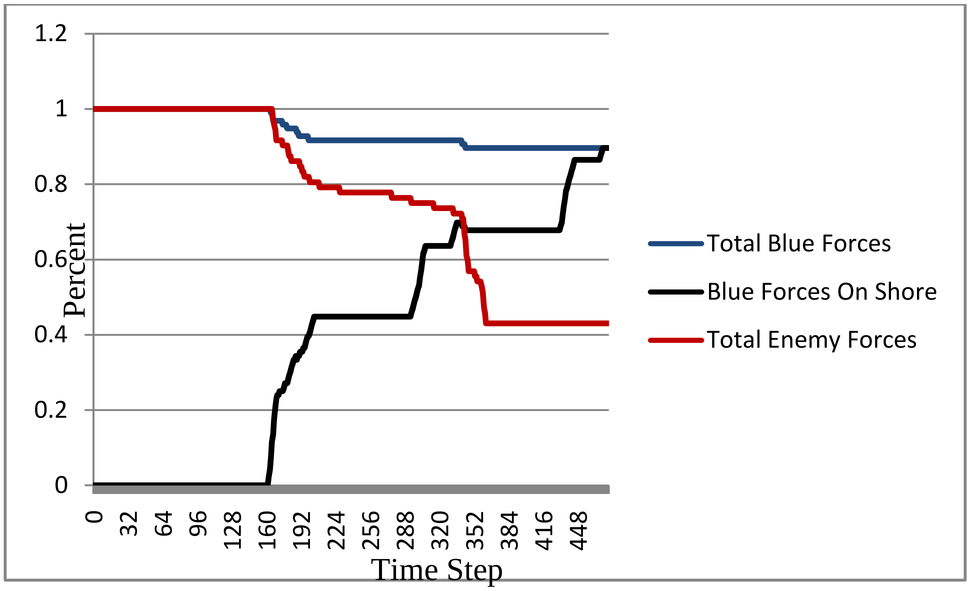

Figure 24. High Speed T-Craft Results

The top graph represents the first scenario (Figure 24). The significant difference is the projected power, or in this case the blue forces on the shore. When the T-Craft are made less capable (Figure 25), there is a drop and lack of rise in the projected power.

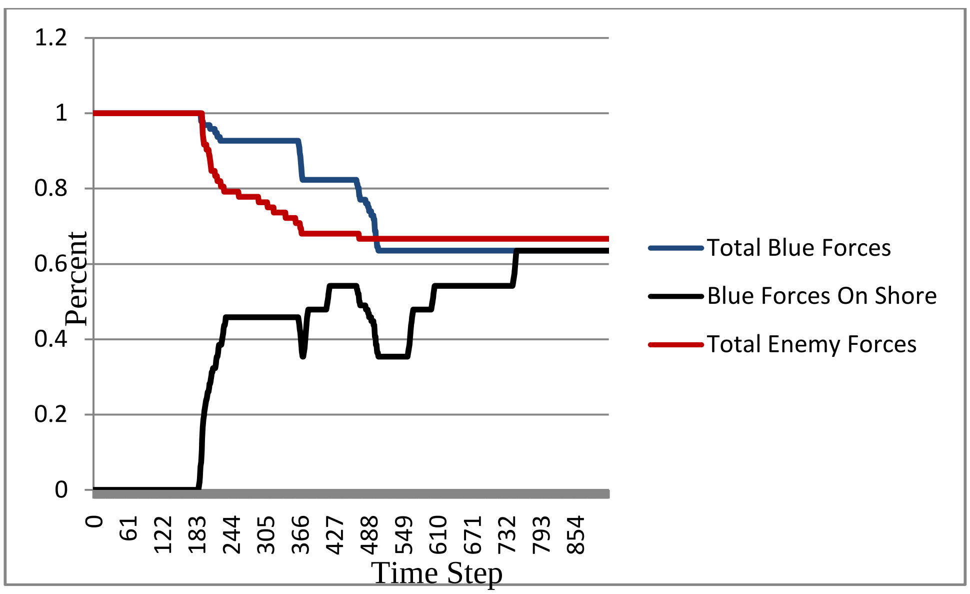

Figure 25. Lower Speed T-Craft Results

It is important to remember that the total number of blue and red forces is kept constant between the two scenarios. In the more successful case, the projected power is not allowed to drop, meaning that more troops are kept on shore reducing the overall casualties. When the blue and black lines meet, this is when all of the blue forces are ashore. In addition, the time it takes for the blue force to fully land on the beach is twice that of the less capable case. These two graphs represent the data that will be analyzed to determine the success of the multitude of cases that will be studied in the following section.

3.4 Automation Setup

A simple Design of Experiments (DoE) was automated and executed on the basic model created in the developed framework shown in the previously. The model consists of red forces that attempt to keep the objectives on the land and blue forces that land troops and attempt to take each of the objectives. The asset of interest in the given model was the T-Craft (Transport Craft). The T-Craft is responsible for moving assets and supplies from ships at sea to the shore and plays a key role in the mission operation. The difference between the two separate T-Craft has already been identified, and using the parameterization and automation, it is possible to vary many more. By varying the parameters outlined in Table 3, a wide range of capabilities can be analyzed while holding the rest of the scenario constant.

Table 3. Ranges of Varied Parameters

Parameters Varied	Minimum Value	Maximum Value
Number of T-Craft in Operation	3	5
Open Sea Speed of T-Craft	10 m/s	30 m/s
Maximum Capacity of T-Craft	300,000 lb	900,000 lb

In addition to the parameter sweep conducted using the small sample DoE, a reliability test can be conducted to ensure that the model does not give random results. There are a large number of factors in the model that rely upon the NetLogo native random number generator. On a second automated test, the number of T-Craft, the maximum weight of the T-Craft, and the speed of the T-Craft were all held constant at 4, 20 m/s, and 600,000 lb respectively. Twenty-seven separate cases were run with these set conditions each with a new random number seed. Varying the random number seed allows the reliability and consistence of the results for a single case to be compared. If the results obtained vary substantially from one another, then there is too much sensitivity to random occurrences.

3.5 Automated Results

The landing profile, or number of troops landed on the shore by friendly forces, is the performance parameter extracted from each DoE run. This is a very similar parameter to the force projection used in single case analysis. The cumulative results from the DoE allow the sensitivity of the T-Craft performance to each of the three chosen parameters to be examined. The first step to show the results that can be produced using this frame work are shown in Figure 26. For each of the varied parameters, the resulting landing profiles can be grouped together.

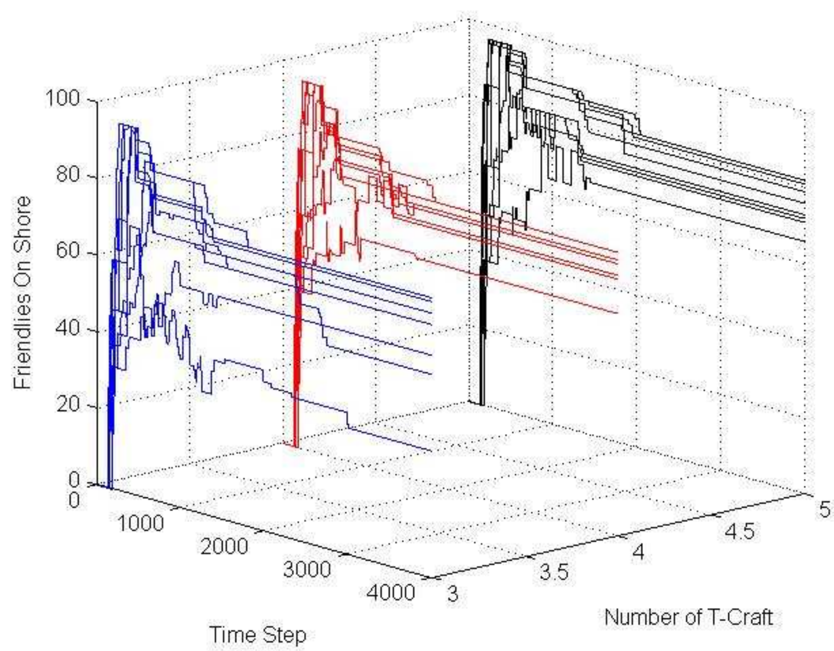

Figure 26. Landing Profiles as a Function of Number of T-Craft

In Figure 26, the resulting profiles have been grouped by the number of T-Craft in each run. In this case, the blue grouping represents three deployed T-Craft, the red grouping represents four deployed T-Craft, and the black grouping represents five T-Craft. The average of each grouping was taken to better compare the differences between them (Figure 27).

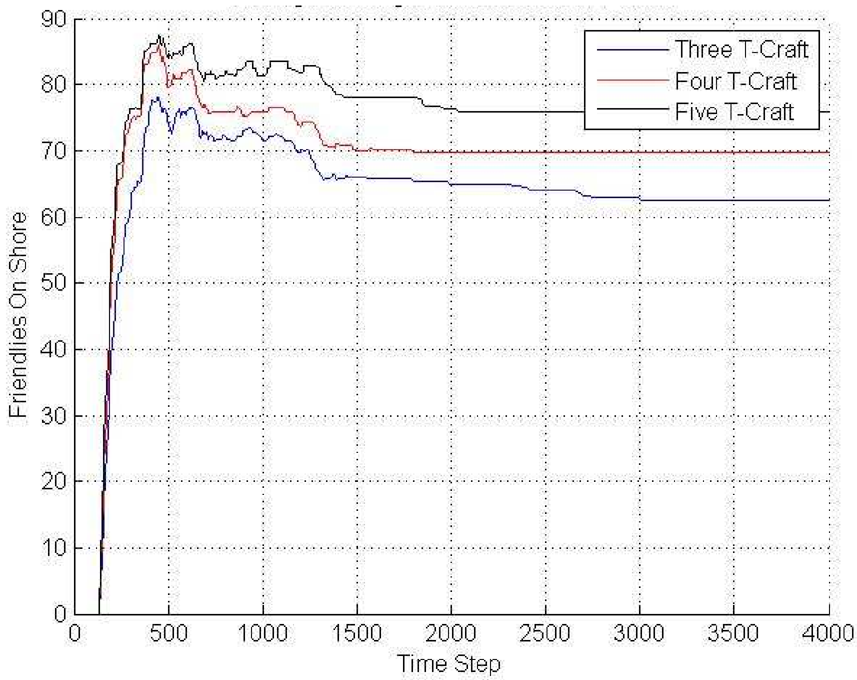

Figure 27. Average Landing Profile: Number of T-Craft

As would be expected, as the number of deployed T-Crafts increases the speed at which the number of friendly troops on shore increases faster. In addition, it can be seen that the more T-Craft that are deployed, the smaller the final casualty count is after the final beach landing. This is shown by the higher final level of troops at the end of the simulations, around 3000 time steps. The more deployed T-Craft, the more favorable the results from the landing operation.

While the number of T-Craft had the expected affect, the landing profile seems to be less sensitive to the maximum speed (Figure 28).

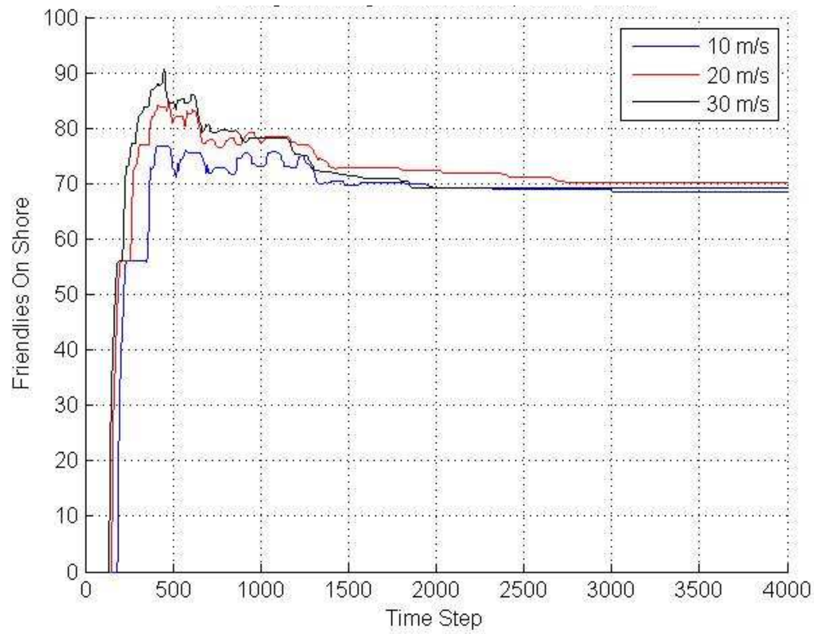

Figure 28. Average Landing Profile: Max Speed of T-Craft

The change in maximum speed of the T-Craft does not have a large affect on the overall performance of the landing operation. A faster speed does allow more troops to reach the beach head faster, but at the same time the resulting casualties, or final troop count on shore, is the same. The effect of the speed on landing rate is much less than that of the number of T-Craft deployed.

The maximum loaded weight, shown in Figure 29, shows an interesting effect. The data collected shows that as the loading capacity of the T-Craft is increased the speed at which forces are landed and the final troop count increases as well. More interestingly, it shows that there are diminishing returns as the weight is increased. At 600,000b and 900,000lb the average landing profiles are very similar.

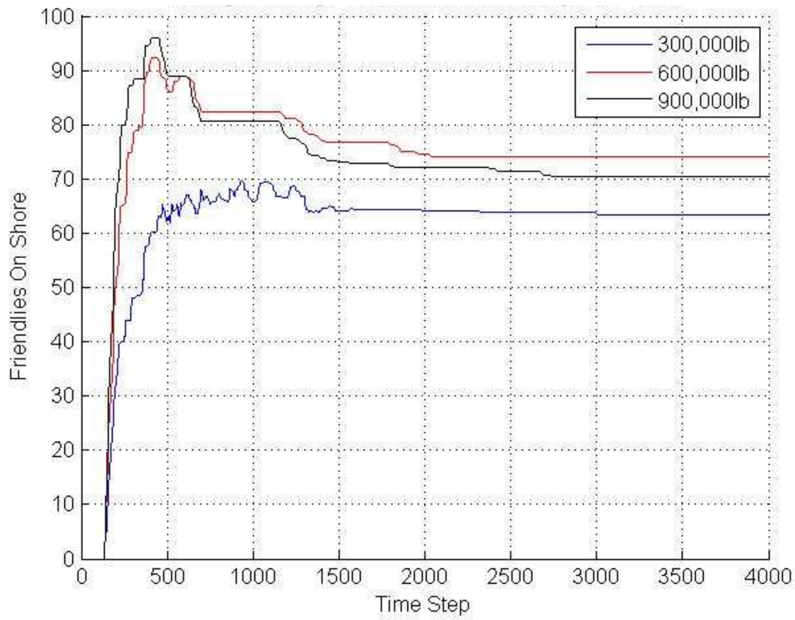

Figure 29. Average Landing Profile: Max Loaded Weight of T-Craft

The sensitivity and effect of the different design parameters allowed the design space to be explored and realized. The framework made this understanding of the complex system possible. The automated capabilities can also be used to check the robustness and reliability of a model created using the framework. To test the reliability and robustness of the T-Craft model, a single case was chosen and run with different random number seeds. The complete results for twenty-seven runs are shown in Figure 30.

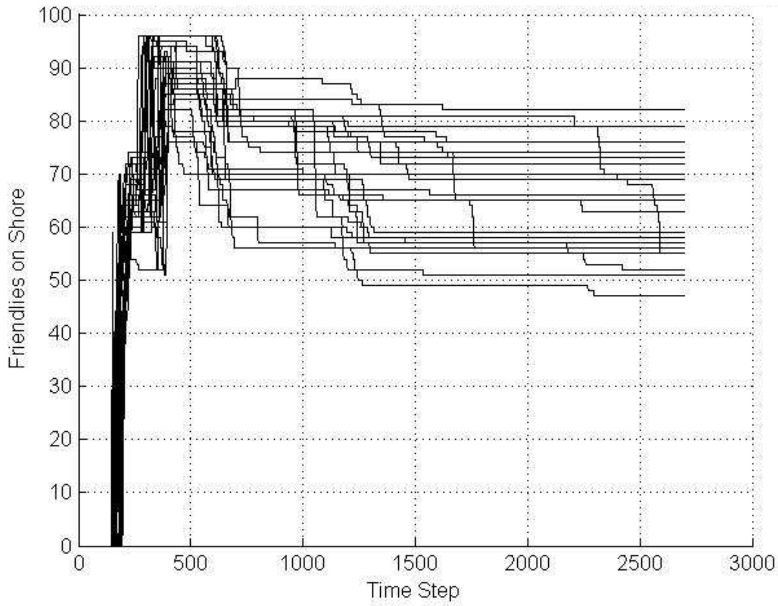

Figure 30. Landing profile for 4 T-Craft with a max loaded weight of 600,000lb and max speed of 20m/s.

Figure 30 shows the variation in the results of a single T-Craft variant run in the same scenario but with a varying random number seed. While the T-Craft is shuttling troops to the shore during the landing portion of the operation, there is little deviation between the separate runs. To demonstrate this average amount landed at a given time has been plotted against all of the individual runs in Figure 31.

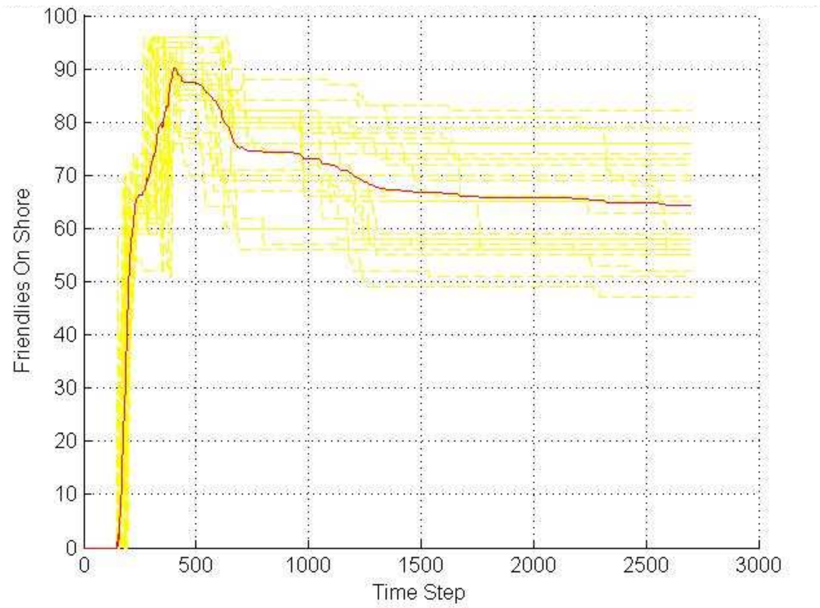

Figure 31. Average landing profile for 4 T-Craft with a loaded weight of 6000,000lb and max speed of 20m/s.

The red line displays the average across all of the time steps. Once troops begin to land, confrontation between the red and blue forces begins and, as a result, a divergence begins to show between the different cases. The divergence means that the combat and altercations once troops begin to engage are heavily dependent on random variables and therefore the random seed. The reliability of the data collected is more reliable and robust for the initial stages of the operation and less so once combat begins. The T-Craft can therefore be evaluated for its landing capabilities using this model, but not necessarily its affect on the combat after landing without sampling a large number of cases. The automation capabilities allowed the results of the T-Craft operation model to be examined and different components verified for reliability.

Chapter IX

Conclusion

The results discussed in Chapter III are a demonstration of the capabilities that the framework possesses. Performance parameters of not only a single asset but an entire system were able to be analyzed as a function of a single asset type. The asset type's capabilities were varied and the effect on the performance of the complex could then be examined through the landing profile. The most important performance parameters for the asset can be chosen from this analysis. The maximum loaded weight of the T-Craft had the largest impact on the scenario as a whole while the number of T-Craft deployed had less of an impact and the speed even less. The evaluation of an assets performance was accomplished because of the frameworks capabilities. In addition, it was shown that the model itself can be examined for reliability and robustness of results.

The model used to demonstrate the capabilities of the framework was created quickly and easily from scratch. This was accomplished because of the already established combat, communication, logistic , and command systems contained within the framework. The ability to automate a large number of runs was due to the lack of human in the loop and the agent based modeling. The framework allowed the model created to be flexibly changed as it needed to be expanded or shrunk. The framework has simplified the process of characterizing both a complex military scenario and the effect of a single asset on that complex system.

Thus success thus far made does not mean that the framework created is in a complete and final form. The framework created is a foundation and stepping stone for future advancement. As the framework is used to create models in the future, more and more plug-ins, or independent components, will be developed and added to the library. These additional plug-ins will be added to those that exist, like the communication or command system. This idea of independent and open advancement is key in the concept behind the development of the framework. Overt time as the framework's use increases, there will be more and more plug-ins available to users beginning to create a model.

There are specific areas that can be concentrated on to maximize the impact of further developments on the framework. First, the combat method can be compared to and calibrated with other models and military data that were not available during the research presented in this thesis. The enhancement of the combat capabilities of the framework would greatly improve the models that can be created. As of now, much of the asset data and combat capabilities must be collected for each model created but, in time, a library of assets and their capabilities would simplify the development process even more. Another high impact area is the expansion of the modeling of the commanders and their decision making capabilities. A more efficient, encompassing, and adaptive commander would allow much more reliable and realistic models to be created using the framework. Current methods to automate the commander include decision trees and triggers, game theory, and trained neural nets.

References

Agents, F. f. I. P. (2002). "FIPA ACL Messsage Structure Specification."

Armstrong, M. J. (2005). "A Stochastic Salvo Model for Naval Surface Combat." Operations Research 53(5): 11.

Beisecker, E. (2008). "Improving Requirements Definition for Systems in the Seabasing Concept through Interactive Visualization." 2008 Grand Challenges in Modeling & Simulation.

Crooks, K. E. and A. Kandel (1992). "Combat modeling using fuzzy expected values." Fuzzy Sets and Systems 47(3): 293-301.

Dupuy, o. T. N. (1985). "Numbers, Predictions, and War: The Use of History to Evaluate and Predict the Outcome of Armed Conflict."

Georgeff, M. P., B. Pell, et al. (1999). The Belief-Desire-Intention Model of Agency. Proceedings of the 5th International Workshop on Intelligent Agents V, Agent Theories, Architectures, and Languages, Springer-Verlag.

Gue, K. R. (2003). "A dynamic distribution model for combat logistics." Comput. Oper. Res. 30(3): 367-381.

Ilachinski, A. (1999). "Towards a Science of Experimental Complexity: An Artificial-Life Approach to Modeling Warfare."

Tiah, Y. M. (2007). "An Analysis of Small Navy Tactics Using A modified Hughs' Salvo Method." Naval Post Graduate School.