A TECHNIQUE FOR DETERMINING VIABLE MILITARY LOGISTICS SUPPORT ALTERNATIVES

A Thesis
Presented to
The Academic Faculty

by

Jesse Stuart Hester

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Aerospace Engineering
Georgia Institute of Technology
May 2009

A TECHNIQUE FOR DETERMINING VIABLE MILITARY LOGISTICS SUPPORT ALTERNATIVES

Approved by:

Prof. Dimitri Mavris, Adviser School of Aerospace Engineering *Georgia Institute of Technology*	Prof. George Vachtsevanos School of Electrical and Computer Engineering *Georgia Institute of Technology*
Prof. Daniel Schrage School of Aerospace Engineering *Georgia Institute of Technology*	Dr. Janel Nixon School of Aerospace Engineering *Georgia Institute of Technology*
Dr. Danielle Soban School of Aerospace Engineering *Georgia Institute of Technology*	Mr. Philip Fahringer Center for Innovation, Suffolk, Virginia Lockheed Martin Corporation

Date Approved: 4 March 2009

In Memory of
My Mother, Victoria & My Father, Tom

ACKNOWLEDGEMENTS

I first would like to thank my entire thesis committee individually. To Prof. Dimitri Mavris, for the opportunities not only given to me here at ASDL, but also his flexibility in allowing me to pursue another degree outside of Aerospace. Nowhere else would I have likely been afforded such an opportunity to personally and professionally develop. To Prof. Daniel Schrage, for lending an ear and making time during his busy schedule to listen to my ideas and progress reports. To Prof. George Vachtsevanos, for his guidance and direction in areas foreign to me and insistence that I bound the problem to a manageable realm. To Dr. Danielle Soban for her advice and input through out the doctoral process. To Dr. Janel Nixon, for her review/editing from the beginning, recommendations into areas that needed more emphasis, and for lending her ear. To Phil Fahringer for his discussions that helped me evolve my thesis topic and ideas in logistics.

Throughout the six years here at ASDL so many people who have had an impact on me by teaching, assisting, and being patient with me when I was in need of further explanation on a subject or approach to a problem. To my fellow classmates, the intense studying and discussion prior to the qualification exams (plural) to enter into the PhD program was instrumental in my learning and helpful in a wide variety of ways; thank you to all who had that impact. For this particular research effort I would be remiss if I did not thank Doug Brown for his assistance with my MatLab questions and in debugging my MatLab GUI. I would also like to thank to Steve Upton, the creator of *Xstudy* and *OldMcData*, for his help. Finally, I would also like to thank the people I communicated with from the Naval Post Graduate School who were very forthcoming with information, data, and software that aided in my experiment executions for this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

AB	Agent Based
ABM	Agent Based Modeling
ABM&S	Agent Based Modeling and Simulation
ANN	Artificial Neural Network
AO	Area of Operation
AOR	Area of Responsibility
AR	Analogical Reasoning
ASD	Agent State Data
ATLAS	Adaptive Technique for Logistics Architecture Solutions
C^2	Command and Control
CBR	Case Based Reasoning
CDF	Cumulative Distribution Function
COA	Course of Action
COTS	Commercial Off the Shelf
DCBR	Dynamic Case Based Reasoning
DM	Decision Maker
DOD	Department of Defense
DO	Distributed Operations
DSS	Decision Support System
DST	Decision Support Tool

EOM	End of Mission
ERP	Enterprise Resource Planning
FIS	Fuzzy Inference System
GDP	Gross Domestic Product
HITL	Human in the Loop
JDDE	Joint Deployment and Distribution Enterprise
JFC	Joint Force Commander
JIT	Just In Time
JO	Joint Operations
JOA	Joint Operational Area
kNN	k Nearest Neighbor
M&S	Modeling and Simulation
MC	Markov Chain
MDP	Markov Decision Processes
MEU	Marine Expeditionary Unit
MOE	Measure of Effectiveness
MSE	Mean Square Error
MTMC	Military Traffic Management Command
MTOE	Modified Table of Organization and Equipment
NCW	Network Centric Warfare
OEC	Overall Evaluation Criterion
OEF	Operation Enduring Freedom
OFT	Office of Force Transformation

OIF	Operation Iraqi Freedom
OMD	Old Mc Data
OMFTS	Operational Maneuver from the Sea
OPLAN	Operations Plan
OPLOGPLN	Operations Logistics Planner
OPORD	Operations Order
PD	Position Data
PDF	Probability Density Function
PHM	Prognostic Health Management
RFID	Radio Frequency Identification
SOS	System of Systems
SSE	Sum Square Error
SST	Software Selection Tool
STOM	Ship-to-Objective-Maneuver
TO	Task Organization
TOE	Table of Organization and Equipment
TRANSCOM	Transportation Command
TSK	Takagi-Sugeno-Kang
US	United States
USA	United States Army
USAF	United States Air Force
USMC	United States Marine Corps
USN	United States Navy

SUMMARY

A look at today's US military will see them operating much beyond the scope of protecting and defending the United States. These operations now consist of, but are not limited to humanitarian aid, disaster relief, peace keeping, and conflict resolution. This broad spectrum of operational environments has necessitated a transformation of the individual military services to a hybrid force that is attempting to leverage the inherent and emerging capabilities and strengths of all those under the umbrella of the Department of Defense (DOD), this concept has been coined Joint Operations.

Supporting Joint Operations requires a new approach to determining a viable military logistics support system. The logistics architecture for these operations has to accommodate scale, time, varied mission objectives, and imperfect information. Compounding the problem is the human in the loop (HITL) decision maker (DM) who is a necessary component for quickly assessing and planning logistics support activities. Past outcomes are not necessarily good indicators of future results, but they can provide a reasonable starting point for planning and prediction of specific needs for future requirements.

Adequately forecasting the necessary logistical support structure and commodities needed for any resource intensive environment has progressed well beyond stable demand assumptions to one in which dynamic and nonlinear environments can be captured with some degree of fidelity and accuracy. While these advances are important, a holistic approach that allows exploration of the operational environment or design space does not exist to guide the military logistician in a methodical way to support military forecasting activities. To bridge this capability gap, a method called Adaptive Technique for Logistics Architecture Solutions (ATLAS) has been developed. This method provides a process that facilitates the use of techniques and tools that filter and provide relevant information to the DM. By doing so, a justifiable course of action (COA) can be determined based on a variety of quantitative and qualitative information available.

This thesis describes and applies the ATLAS method to a notional military scenario that involves the Navy concept of Seabasing and the Marine Corps concept of Distributed Operations applied to a platoon sized element. The small force is tasked to conduct deterrence and combat operations over a seven day period. This work uses modeling and simulation to incorporate expert opinion and knowledge of military operations, dynamic reasoning methods, and certainty analysis to create a decisions support system (DSS) that can be used to provide the DM an enhanced view of the logistics environment and uses variables that impact specific measures of effectiveness. The results from applying the ATLAS method provide a better understanding and ability for the DM to conduct the logistics planning / execution more efficiently and quickly. This is accomplished by providing relevant data that can be applied to perform dynamic forecasting activities for the platoon and aids in determining the necessary support architecture to fulfill the forecasted need.

1 INTRODUCTION

The effects of globalization have had significant and unforeseen effects across the world in such a way that people, businesses, and countries are now integrally connected and dependant on each other in a variety of ways. A well know economist, Thomas Friedman, identifies in his book, *The World is Flat*, ten enablers that he considers the catalysts of globalization. The first two he contends are the falling of the Berlin wall and the emergence of Netscape, the medium that kick started internet accessibility to the common user in 1995 [36]. Arguably, the second is the most profound and extensive happening in the 20th century. Exponential advances and uses of internet technology in everyday life have provided enhanced visibility, increased efficiency, and vastly increased the amount of information available to the average person. This has allowed anyone with a network or internet connection the ability to gather large amounts of information and data for what ever reason they need to. While beneficial at times, it has also made decision making much more cumbersome. This is especially apparent for leaders and other decision makers (DM) across a variety of industries. For many DM's, management of resources and timely delivery of their commodities is critical to their organization's success. The ability to forecast or predict the needs of their customers and getting the right resource delivered to the right place and at the right time is paramount. During the 1980's and early 1990's, in order to gain a competitive advantage, one only needed to implement technology and data management processes that were the result of exponential advances in computing technology. Now gaining a competitive advantage is much more difficult and requires not only the latest computing applications, but also new and innovative advancements in the realm of information and data management techniques in order to realize increased efficiencies and capabilities. The following chapter will provide an overview of a world wide organization that is seeking to enhance its knowledge and perform logistics and resource management in a more efficient manner; the United States military.

1.1 Motivation

"Amateurs talk strategy. Professionals talk logistics."

-Unknown

Logistics has always been a thorn in a commander's side. A look at just about any military campaign in history will provide justification for this statement. The Romans it is surmised developed and mastered the art of logistics during their expansion of power and dabbling into conquering neighboring lands, particularly Sicily and battles in the First Punic War [39]. Since then, logistics has continued to have an ever increasing role in any military's architecture. Napoleon's success and downfall were the result of his well managed supply lines and lack there of. By June 1812, Napoleon had virtually all of continental Europe under his control. When Napoleon decided to attack Russia, his available forces consisted of 500,000 soldiers. Because of the advantage was clearly in the French commander's favor, the Russians decided to avoid direct confrontation with the French Army. This drew Napoleon further and further into Russia as the summer wore on. Consequently, Napoleon's massive supply lines stretched thinner and thinner over the months and his forces declined in all facets due to lack of adequate supplies and resources [93]. By September of that same year, without having engaged in a single major battle, the French Army had been reduced by more than two thirds of its original size from fatigue, hunger, desertion, and guerilla raids by Russian forces [93].

Today, with recent military involvement in the Middle East and Asia, logistics has continued to play a pivotal role in determining the outcome of military action in the air and on the ground. Desert Storm, the first Gulf War, was over in 5 days from start to finish, but it took months to preposition and prepare for the offensive. Some say it would have been over sooner had it not been the inadequate logistics supply system that slowed the speed of advancing forces [102]. In the first 30 days of the buildup for Desert Storm, called Desert Shield, 38,000 troops and 163,581 tons of equipment were landed and processed [102]. Over the course of the entire conflict, items processed on average per day were 35 planes and 2.1 seafaring vessels; in all consisting of 33,100 containers offloaded; equal to 188 miles if laid end to end [102]. Operations for today's military require a flexible, dynamic, and robust support network capable of getting resources where they need to be in an efficient and timely manner.

There are many factors to consider when determining force composition and the ensuing support architecture for any operation. These factors entail components like strategy, tactics, and the operational end goal when the planning and operational phases are conducted, but logistics is a major factor, if not "the key" component, to promoting victory. Recently in September of 2007, a United States Transportation Command (TRANSCOM) presentation stated that in the area of deployable distribution command and control, the primary task should be to:

> Develop a Joint Logistics Evaluation Model to assist the Joint Force Commander (JFC) in the selection of the optimum Joint Logistics Management Option and other selected Logistic Capabilities (to include Distribution) to best support operational requirements;

with a final product outcome consisting of a "Joint Logistics Evaluation Model that can be utilized to support Joint Logistic Capability decisions" [135]. The reader can start to see from this statement that significant emphasis is now being placed on developing a logistics support framework that is robust and capable of supporting the evolving military's needs for combined (multi-service) operations.

1.1.1 Military Logistics Defined

The word logistics is defined as the science of managing and controlling the flow of goods, energy, information and other resources like products, services, and people from the source of production to the marketplace [71]. As such, logistics is commonly considered a branch of engineering which creates "people systems" rather than machine systems. Logistics planning has always been a necessity for the military, especially when a vast amount of resources is necessary to achieve a goal. For the professional soldier, military historian, and or military theorist, accurately and unequivocally agreeing on what the word *military logistics* entails is difficult [131]. Even today, the meaning of logistics can be somewhat fuzzy in spite of its frequent usage and reference in DOD publications and the existence of lengthy definitions in various service and joint regulations. Historian Stanley Falk, in the author's opinion, best describes logistics on two levels. First, at the intermediate level:

> Logistics is essentially moving, supplying, and maintaining military forces. It is basic to the ability of armies, fleets, and air forces to operate--indeed to exist. It involves men and materiel, transportation, quarters, depots, communications, evacuation and hospitalization, personnel replacement, service, and administration [45].

Second, at a higher level, logistics is:

> ... economics of warfare, including industrial mobilization; research and development; funding procurement; recruitment and training; testing; and in effect, practically everything related to military activities besides strategy and tactics [45].

While there are certainly other definitions of logistics, Falk's encompassing definition and approach provides an ideal backdrop from which to examine and discuss *military logistics*.

From a historical perspective, ten major themes have been identified that categorize modern United States (US) *military logistics*. Observations that give support for the focus of this thesis topic are underlined [119]:

- The tendency to neglect logistics in peace time and expand hastily to respond to military situations or conflict.

- The increasing importance of logistics in terms of strategy and tactics. Since the turn of the century, logistical considerations increasingly have dominated both the formulation and execution of strategy and the tactics.

- The growth in both complexity and scale of logistics in the 20th century. Rapid advances in technology and the speed and lethality associated with modern warfare have increased both the complexity and scale of logistics support.

- The need for cooperative logistics to support allied or coalition warfare. Virtually every war involving US forces since World War I has involved providing or, in some cases, receiving logistics support from allies or coalition partners. In peacetime, there has been an increasing reliance on host-nation support and burden sharing.

- Increased specialization in logistics. The demands of modern warfare have increased the level of specialization among support forces.

- The growing tooth-to-tail ratio and logistics footprint issues associated with modern warfare. Modern, complex, mechanized, and technologically sophisticated military forces, capable of operating in every conceivable worldwide environment, <u>require that a significant portion, if not the majority of it, be dedicated to providing logistics support to a relatively small operational component</u>. At odds with this is <u>the need to reduce the logistics footprint in order to achieve the rapid projection of military power</u>.

- The increasing number of civilians needed to provide adequate logistics support to military forces. Two subthemes dominate this area: first, unlike the first half of the 20th century, less reliance on the use of uniformed military logistics personnel and, second, the increasing importance of civilians in senior management positions.

- The <u>centralization of logistics planning functions and a parallel effort to increase efficiency by organizing along functional rather than commodity lines</u>.

- The application of civilian business processes and just-in-time delivery principles, coupled with the <u>elimination of large stocks of spares</u>.

- <u>Competitive sourcing and privatization initiatives that replace traditional military logistics support</u> with support from the private business sector.

As can be seen from above, the importance of logistics and the proper management and selection of the support structure that evolves within and or outside the military are topics that are gaining significant attention. If the military is to transform itself to meet the evolving operational environment, leveraging techniques and methodologies that will make determining how to best support the force will be critical to facilitating a robust support network capable of dealing with the increasing asymmetric threats that are increasingly appearing across the operational landscape.

1.1.2 The US Military and its Role in the World

An entity like the US military has to be able to operate anywhere in the world. To do this they must be able to deliver and receive any necessary supplies quickly and efficiently. Finding efficient ways to forecast the needs for this complex dynamic operational environment is a challenging task and is one that continues to be researched and investigated by many groups in the government and the private sector. These groups consist of, but are not limited to the Army and Naval War Colleges, the Brooking Institute, the American Enterprise Institute, and the Heritage Foundation. The end of the Cold War and the increasing emergence of third world and regional adversaries have prompted the US military to reassess the architecture of its military in order to meet this changing operational landscape [103]. In the past, the military's logistics architecture was designed for its core mission; to plan, train, and prepare for large scale combat operations with well defined adversaries. These large scale operations tended to allow the US military to utilize overwhelming force, superior fire power, and its technological advantage to close with, fix, and defeat an enemy; that has all changed. In November of 2007 the Army Chief of Staff, Gen. George Casey stated three observations facing the current and future military force [38].

- Global terrorism and extremist ideologies are a reality. The next decades will be one of persistent conflict. This is a period of protracted confrontation among states, non-state, and individual actors who are increasingly willing to use violence to accomplish their political and ideological ends.

- Globalization is a trend that exacerbates protracted confrontation. It has also created 'have' and 'have-not' conditions that are ripe for exploitation.

- Technology is another double-edged sword. The same innovations that improve quality of life and education and livelihood are also used by extremists to export terror around the globe and manipulate our media.

Today, the operational environment has transitioned away from large scale operations to missions where much smaller forces such as platoon or company sized elements, are more effective in dealing with a threat or problem [103]. The threats facing the United States today have become increasingly ambiguous as non-state actors attempt to utilize irregular tactics to achieve their goals via catastrophic or disruptive methods [116]. This environment necessitates a rapidly deployable force and thus a logistics architecture that is both robust and dynamic in terms of capabilities.

The goal of the Department of Defense (DOD) is to develop and have at its disposal forces that are agile, capable, and robust enough to quickly close with and decisively eliminate threats [128], [129], [130]. Joint Operations (JO) is defined as operations conducted by two or more branches of the service working together. These joint ventures have become common place in today's military operational environment. The primary goal of these partnerships is to effectively leverage the strengths of each service and effectively utilize any emergent capabilities that appear and promote accomplishing a mission. Increasingly, JO's are being conducted in an effort to deal with the increasingly fragmented and complex battle space that is increasingly becoming filled with asymmetric aggressors [115].

A sought after byproduct of JO by military leaders is the ability to effectively create confusion for any adversary by outwardly appearing as a disjointed and fragmented force that in reality, is capable of rapidly reconfiguring itself to perform a variety of operations [116]. Success in this endeavor should make it harder for an enemy to predict the actions of the forces in theatre or allow the enemy to react with any efficiency [115]. While the military is currently adapting to joint operations "on the fly" they are also attempting to go through a large transformation in their physical force architecture. This transformation consists of but is not limited to the incorporation of new technologies, identifying/filling security gaps, and reconstituting or fielding of new equipment [59].

As these changes ensue, senior military leaders have realized that a holistic approach, one that encompasses the needs of all the armed services and leverages new technologies from the bottom up is the way of the future and must be pursued [57]. This leadership is fostering a command climate conducive to facilitating integration of the individual services, but faces hurdles in the form of inadequate infrastructure for such a dynamic force as well as buy-in from the individual services that a viable integration of the services can become a reality. Several gaps exist in the area of providing a network centric system capable of rapidly assessing and facilitating decisions of how to best employ and sustain the small unit forces envisioned for the future [58]. These gaps are further widened because the DOD has placed the constraint that the bulk of these operations for the foreseeable future will originate and operate in the *littorals*, i.e. the area of land and sea nearest a coastline [133].

For the Navy, this directive shifts their focus closer to shore rather than their historical focus - the open seas. For the Marines and Army ground forces, the focus on the *littorals* will require their conventional large units to be reorganized into reconfigurable smaller forces. Each service has attempted to identify the right capabilities mix necessary to operate in this new environment. The US Army (USA) began a transformation in the late 90's under General Shinseki towards a concept called the Stryker Brigades. The general mission for this Army unit type would be to deploy globally with a mission tailored force package of combat, combat support, and combat service support forces to conduct strategically responsive operations in support of JO contingencies [43]. They would be capable of commitment across the full spectrum of conflict and be optimized to conduct small-scale contingency operations and to deter or contain crises by employing the full range of Army, joint, multinational, and interagency capabilities [43].

The US Marine Corps (USMC) investigated and continues to pursue a concept called Distributed Operations (DO) [32], [33]. Initial experimentation of this concept consisted of a platoon separated into two command teams and three squads with a variety of weapons and communications assets, and measured the resultant combat capabilities [24]. The goal of this experiment was to spur the spiral development of viable low-level force architectures that could be applied to larger force compositions for future scaled deployments of the military. Distributed Operations is meant to constitute a form of maneuver warfare by utilizing small, highly capable units spread across a large area of operations. The USMC believes with this concept, they will experience a spatial advantage commonly sought in maneuver warfare, in that they will be able to sense an expanded battle space and be able to use close combat or supporting arms, including joint fires, to disrupt the enemy's access to key terrain and avenues of approach [31].

The US Navy (USN) focus on the littoral has resulted in a concept called Sea Power 21. This concept brings together the effects of several sub-concepts to provide force protection, intelligence, surveillance, reconnaissance, and logistics [115]. Core to this concept is Seabasing. Seabasing is the idea of using ships to place expeditionary airfields and ports at sea in regions where it may not be geographically or politically feasible to have a large contingent of forces ashore [9]. It has been described as the having capabilities of "the teeth ashore and the tail afloat" by the Joint Chiefs of Staff in 2005 [9]. Although conceived by the Navy, Seabasing is being embraced by the Army and Marine Corps since the concept is meant to utilize ships as floating disembarkation and logistical supply bases for deployed ground forces. With these ships and their organic or supplemental aerial assets, the USMC and USA believe they can become more flexible and rapidly deployable, while substantially reduce their logistics foot-print ashore [116]. A brief overview of Seabasing and DO will be discussed in Section 1.1.3 and 1.1.4 respectively.

1.1.3 Seabasing

The National Security Strategy, National Defense Strategy, and National Military Strategy as outlined in current doctrine all emphasize the need to retain global freedom of action without the need for host nation support in a potential operational area [128], [129], [130]. To accomplish this strategy and in an attempt to meet the demanding operational tempo facing the services, several initiatives have been formulated. The primary solution being pursued by the Navy is a concept called Seabasing under and overarching initiative called Sea Power 21. This concept is envisioned to be implemented between the 2015 – 2025 time-frame. Figure 1 shows the current architecture overview of how a Seabase will be integrated into the intra theatre logistical system.

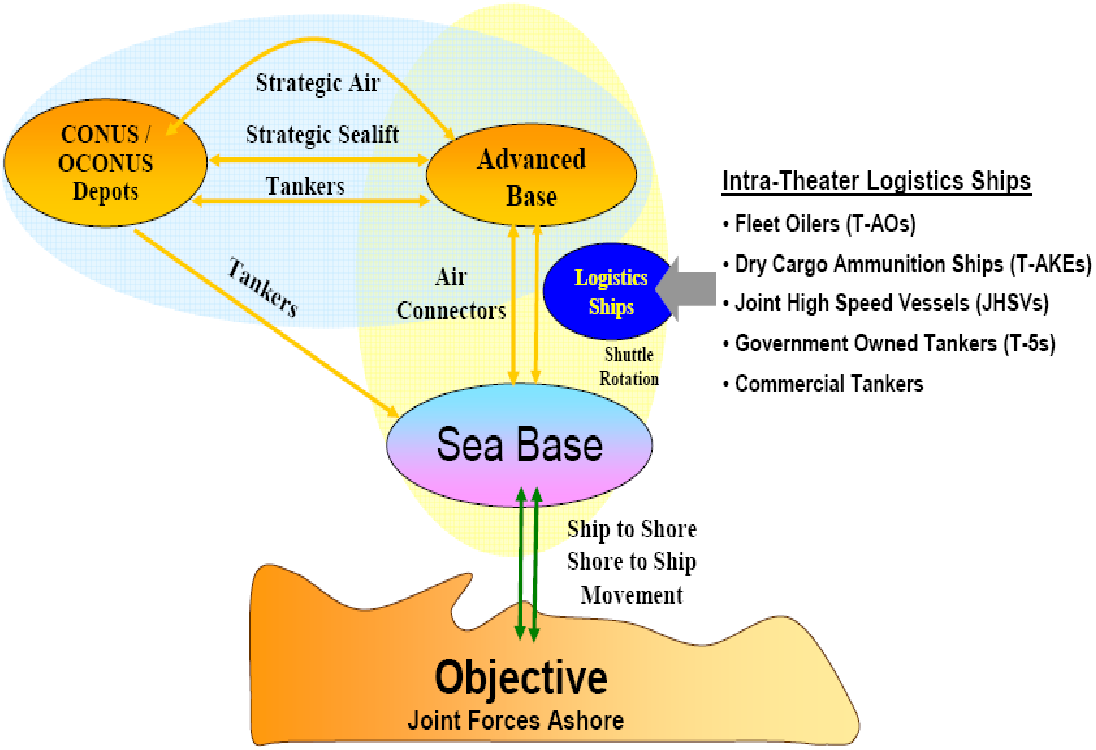


Figure 1: Seabasing Logistics Architecture Overview [59]

The Seabase concept is designed to leverage U.S. Navy forces already in a region by joining ships together a tailored force capable of delivering troops, tanks, and other combat power and then sustaining those forces logistically. Seabasing will also provide support by fire capabilities and safe havens for reconstituting or retrograde options for these combat elements. Seabasing is ideologically defined as the "rapid deployment, assembly, command, projection, reconstitution, and re-employment of Joint combat power from the sea, while providing continuous support, sustainment, and force projection to select expeditionary joint forces without reliance on land bases within the Joint Operating Area (JOA)" [58]. Achieving this ability will require modifications to some current ship, aircraft, and vehicle components throughout the Armed Forces architecture, but the real challenge will be creating a logistical system capable of supporting the diverse (multi branch) and distributed force that will be needed for a variety of uncertain operational environments.

Seabasing

Speed, Access and Persistence

Using these principles

The sea as maneuver space…

Leverage forward presence…

Expand access options

… reduce dependence on land bases

Create uncertainty…

Protect joint forces…

Scalable, responsive joint power projection…

Sustain joint force operations…

Seabasing will be

Interoperable and interdependent

Fires – NSFS, Aviation, Bombers

Maneuver - Connectors

Sustainment – Timely, efficient

C4ISR – Common Operating Picture

Force Protection – Air, Sea and Land

…and will allow us to pursue these lines of operation

Close: Within 10-14 days

Assemble: At sea within 72 hours of arrival

Employ: One Brigade within 8-10 hrs

Sustain: Up to two joint brigades

Reconstitute: For follow-on operations

Figure 2: Aspects and Objectives of Seabasing [116]

As can be seen in Figure 2, the Seabasing concept is envisioned as the key element that will facilitate a dynamic logistical support structure for asymmetric operational environments vicinity the *littorals*. According to Marine Brig. Gen. Bruce Byrum of the Navy’s expeditionary warfare division:

> Seabasing is not something new. We’ve been doing it. We’ve been doing it jointly. But basically it’s been ad hoc. Our forces are not necessarily organized or designed to work efficiently together, and we’re trying to work on that deficiency [9].

The sea base will be the primary source of logistics support for the task force commander and it will also be the conduit through which supplies, equipment, and personnel will move ashore for reinforcement and sustainment operations as well as the source for non-end-item user maintenance, engineering, and health service support to the force [42].

1.1.4 Distributed Operations (DO)

The concept of *Distributed Operations* is described as an operating approach that will create an advantage over an adversary through the deliberate use of separation and coordinated, interdependent, tactical actions [31]. DO builds on the concepts of Operational Maneuver from the Sea (OMTFS) and Ship to Objective Maneuver (STOM) [40]. The essence of this concept centers on the ability for forces to conduct coordinated action, throughout the breadth and depth of the battle space they occupy [31]. Figure 3 conceptualizes this ability and shows how enhanced awareness and communication can

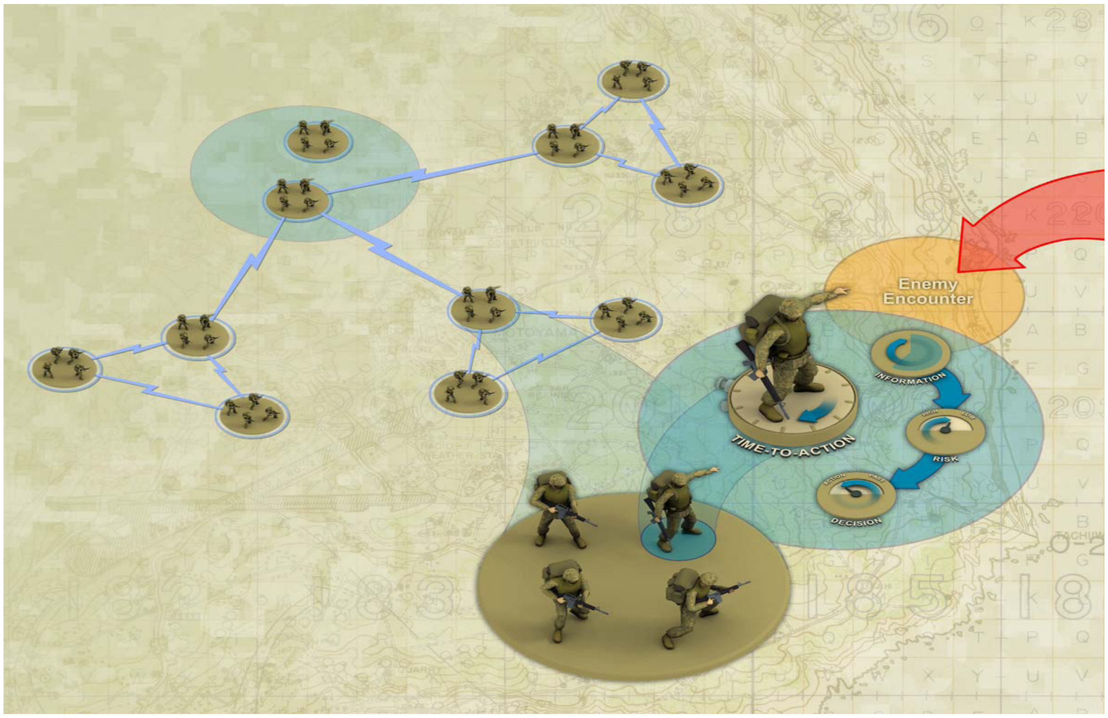


Figure 3: Combat Decision Makers Environment [31]

facilitate rapid reconfiguration of forces to be able to react to a situation. This concept is meant to increase access to various support entities through communication and technology applications, as well as enhances combat capabilities at the small-unit level via direct support from nearby sea assets. With this concept, unit commanders on the ground will be allowed freedom of action to disperse their forces or aggregate them as the situation dictates. While the exact size and makeup of these forces is still being debated, operating in a fractured and ill defined operational environment of the future will require effective command and control that is enabled by the DO concept. Determining the best way to logistically support of these forces appears to be the primary limitation [9].

In the past, operating in ambiguous and volatile environments consisted of deep patrols with units generally sustaining themselves with what they could carry or acquire in the form of food, water, ammunition, etc. [133]. This was a marginally easy task as the goals of these deep operations were primarily intelligence gathering, reconnaissance for follow on operations, and surveillance. The future tactical unit force will now be faced with the full spectrum of military operations from humanitarian aid to full combat conditions [33], [40], [42]. Robustness of the force is paramount and will likely involve a spiral development to determine the right architecture and adequate complementary support network to enable operating in these environments.

1.1.5 Seabasing and DO Together

Addressing the issue of timely, accurate, and efficient logistical support for US forces under Seabasing and DO requires a few assumptions [127].

- First, these forces will be and for the foreseeable future continue to operate in a joint capacity.

- Second, rapid infusion of reliable and suitable technology will be fielded as soon as possible to facilitate these operations.
- Third, the military will continue to be utilized as a tool to enforce national and international policy outlined by leaders in the United States and its allies.
- Finally, there exists no near-term peer competitor likely to emerge to challenge United States access to the common use of sea, air, or space.

The last one is critical to the concept of the sea base because it assumes that off shore ships will be relatively safe from attack by the irregular forces the US military will likely face in the developing operational environment.

As alluded to earlier, the goal sought by the DOD is to increase operational capability and agility by offering increased combat capability through efficient architecture and selection of the best course of action (COA). The COA selected should capitalize on joint capabilities that can provide increased survivability and sustainability. Since there exist a variety of operational environments, varying force architectures will be needed for deployment. The definition of architecture for the purposes of this problem is defined as the composition of a joint force in terms of physical assets. These assets will likely consist of a variety of ships, aviation and ground units, and other force components organic or added to the overall force created to accomplish an assigned mission. A COA is defined as any action or potential action under consideration for the purposes of accomplishing the mission at hand. A COA is directly correlated to forecasting the amount of logistics needed to perform a given mission or scenario.

In order for the concept of Seabasing and DO to succeed an effective command and control network which facilitates information sharing and operational space visualization is necessary to support the various forces spread across the operational area. By being able to sense and visualize the environment, the thought is the DM will be able to plan and respond appropriately. This network will hinge on the ability to provide coordinated logistics support for forces in the battle space effectively. In effect, the operational network employed must be able to *sense and respond* to the logistical needs of those involved. Further discussion of the concept of *sense and respond* in terms of military operations will be discussed in Section 1.3.2

1.1.6 Joint Operations and Ramifications for the Military

In the past large scale military operations operated on supply based logistics, meaning that a local or regional supply base was utilized to sustain military forces in the area. That concept is now changing to distribution based operations to accommodate smaller forces that will tend to be more decentralized and unique in their needs. In 2003 Major General Ann E. Dunwoody, the commander of Military Traffic Management Command (MTMC) stated that "the shift from supply-based to distribution-based logistics is imperative to any successful future military operations" [47]. The transformation is requiring the MTMC to go through not just organizational changes, but process and cultural changes as well. Their end goal is to have one organization that all war fighting allies can turn to for all of their transportation needs; whether it is by land, sea, or air [47]. By pursuing this goal, enhanced visibility over all aspects of the logistics line from beginning to end should be realized. Many recognize that this goal has and will continue to be an evolutionary one, with really no end state as a result [47]. This realization is born from the observation that as the military structure continues to evolve, so must the organizations that provide its logistical support.

What the military primarily hopes to see in this spiral effort is significant compression in response time, improved service, and reduced required investment in current inventories across all aspects of the supply chain network [47]. In order to do this, the logistical support structure now must become a very nimble and dynamic system, capable of predicting and forecasting the needs for any type or size force the military can come up with.

1.2 Current Environment for the US Military

The US military as a whole has been reducing in numbers over the years, yet experiencing increasing requirements to be able to operate on a global scale. This necessitates a streamlined, fluid, and flexible logistics network capable of adapting to the needs of a variety of military units [30]. Figure 4 depicts the current breakdown of the globe and the regional areas of responsibility (AOR) between the major US commands.

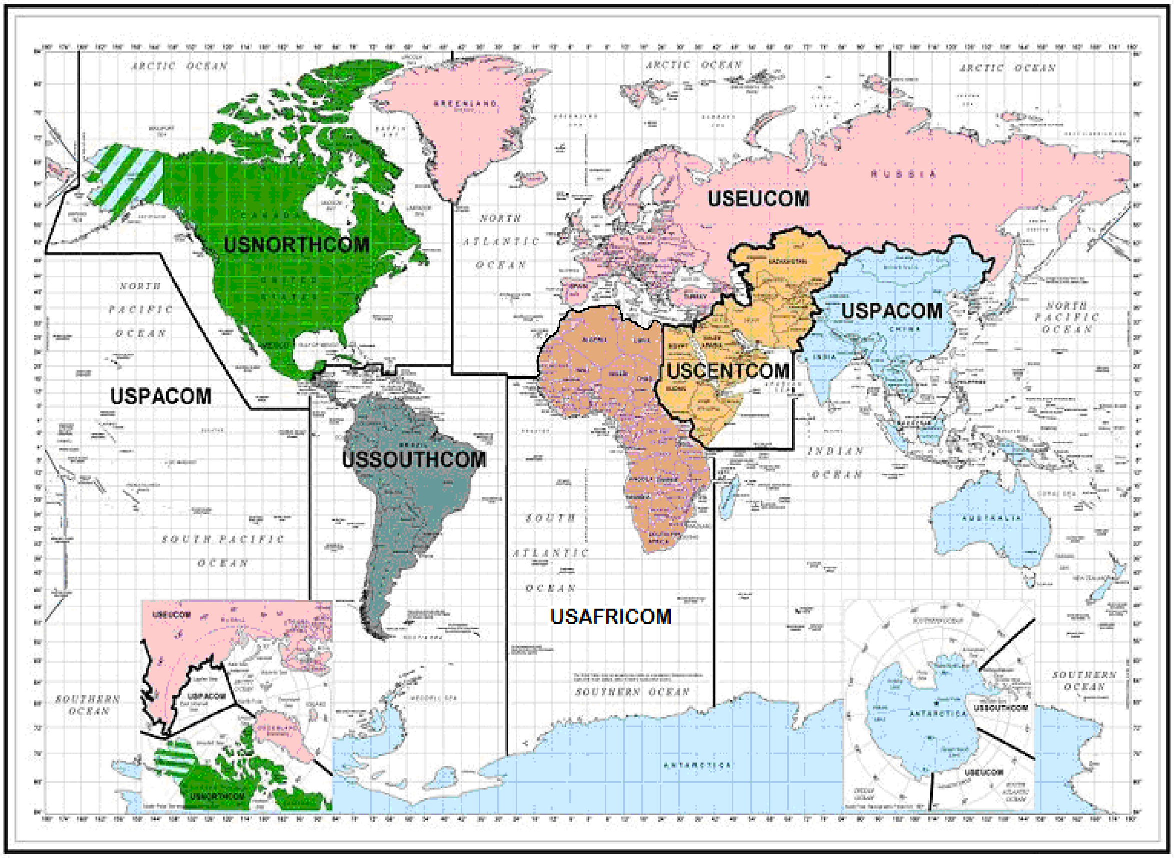


Figure 4: The World with US Commanders Area of Responsibility [29]

The commander for each of these regions, seven in all, will not only need to know what is happening in their AOR, but must also consider events and information about operations out of their AOR so they can holistically factor these aspects into their decision making process. Resources in terms of ships, soldiers, and other assets are not an endless supply. This requires innovative and efficient management activities from leaders at the top all the way down to the operational command and control level.

The subsequent pair of figures reflects the allocation of funds for US defense budget characterized two different ways. The defense budget includes, but is not limited to line items like research and development, paying the force, training, force protection, and operational expenditures [98]. Figure 5 shows the relative expenditure of funds toward the national defense budget as a percentage of Gross Domestic Product (GDP).

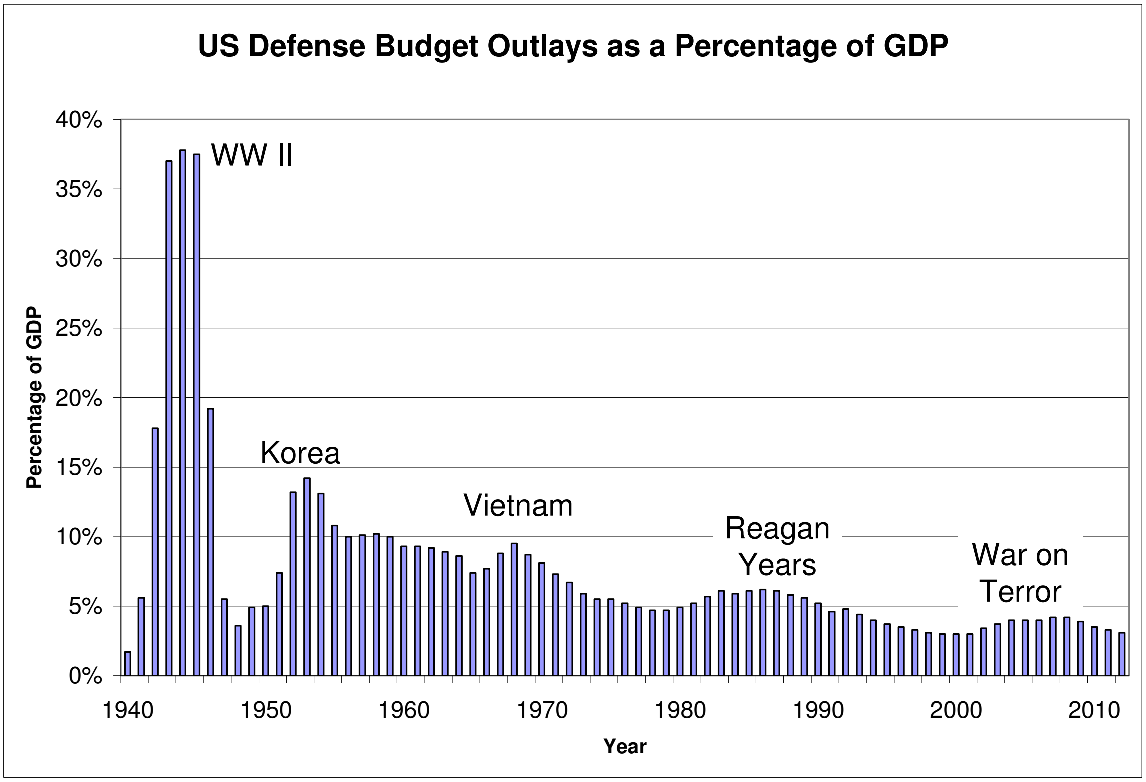


Figure 5: US National Defense Outlays as a Share of GDP [98]

As can be seen the relative spending as compared to GDP has decreased significantly since World War II with an experienced increase around each major conflict the US has been a part of, but the trend is it has continued to fall as time has progressed. Some may argue that although there is a decrease in the percentage of GDP spent on defense that the budget is adequate when adjusted for the growth in GDP the US experienced during these years. Figure 6, below, depicts the actual dollars spent in 2003 dollars, with the years 2007 through 2012 forecasted by the Office of Management and Budget [98]. As can be seen funding has remained somewhere between $300 to $500 billion with an increase projected over the next couple of years and then a forecasted reduction after 2008.

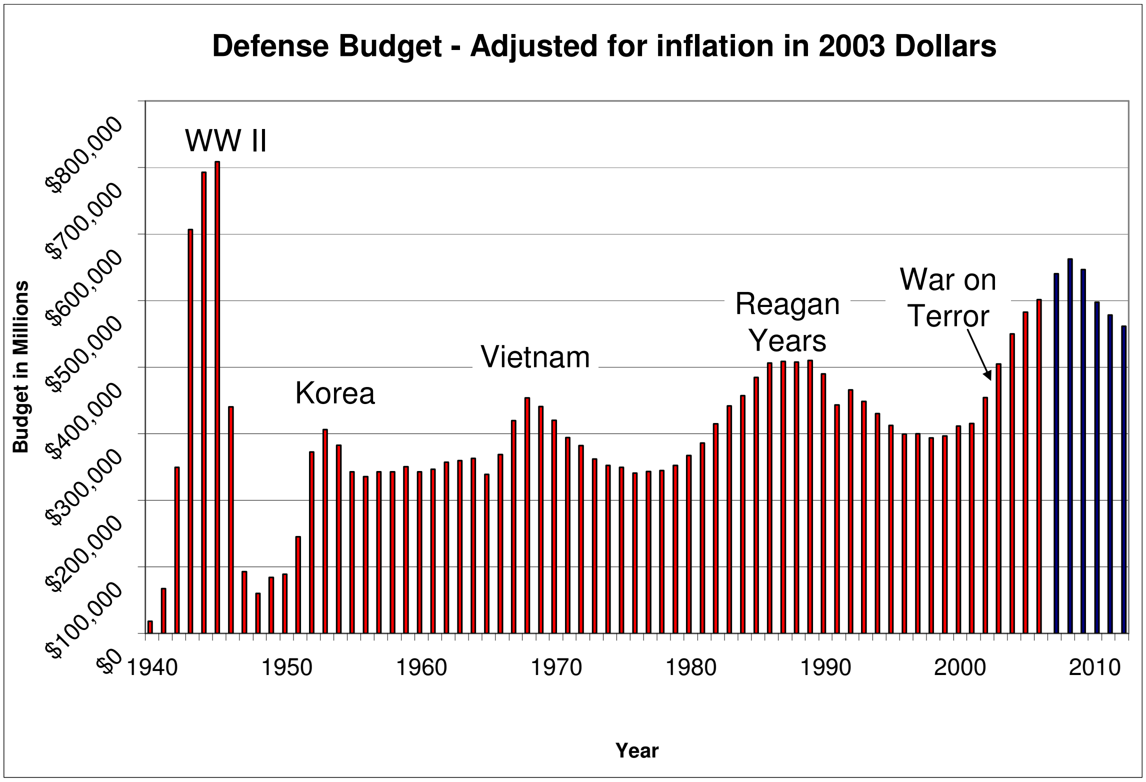


Figure 6: Defense Budget 1940-2012 in 2003 dollars

What is poignantly clear from Figure 4 through Figure 6 and current military conditions is that:

- Operational responsibility for the US military has expanded to cover not just the US, but the entire globe with significant activity currently occurring in the Middle East and Asia.
- The defense budget continues to get less and less in terms of the GDP allocation for its fiscal budget.
- The military although asked to increase its roles and responsibility across the globe, is expected to do so at relatively the same budget since the mid to late 80's, even though the amount of responsibility has increased substantially.

The expectation of the US military by civilian leaders is for the armed forces to be able to react whenever, wherever, and to whatever they are called upon. Being able to do this efficiently and quickly is the daunting task facing US military community. Up until now, this has been accomplished by tapping into experience and operational know how of senior officers and soldiers within the military structure. Currently, there exists no repository of historical information, lessons learned, or knowledge gained from past experiences that can be quantitatively or qualitatively used to guide the future military DM's in the ensuing operational environments. To address this gap, a concept called Network Centric Warfare (NCW) has been proposed and articulates many top level initiatives intent on solving this problem. While not desiring to expressly tell the military how to conduct its current and impending transformation, this concept aims to provide insight from a bottom up perspective for senior military leaders on how to best approach growing requirements holistically.

1.3 Network Centric Warfare

The United States, as well as its allies must be able to face the emerging asymmetric threats at the global, regional, and or local level. Emerging aggressors are non-state, multi-dimensional, flexible, distributed, and information-aware [100]. They have shown via recent events that they are capable of rapidly adapting to and neutralizing U.S. strategies and tactics. Increasingly, these entities have at their disposal catastrophic, inexpensive, and competitive methods of creating large-scale disruption and or destruction [100]. With these facts in mind, the US military is continually exploring how to tackle these multifaceted problems. In October of 2001 the Office of Force Transformation (OFT) was created to help change the military and its culture from the bottom up through a combination of large amounts of experimentation, increased sharing of new knowledge and experiences, and ideas to broaden military capabilities while mitigating risk [97]. To accomplish this task the OFT put forth one of its pillar theories as a guide to military leaders called Network Centric Warfare.

Historically, military logistics has taken a mass-based approach to satisfying operational needs, creating what is commonly referred to as *Iron Mountains* to supply itself. This consisted of one or a few strategically placed and often massive stockpiles that had its size and composition determined by a metric of days of supply per person for consumables and other daily measures from aggregated historical data. This application was seen most recently in the Persian Gulf War, described earlier, where the build up began in August of 1990 for an operation that took place in late February 1991. Mass based logistics may still continue to work for large operational conflicts and if the situation permits time for logistical build-up, but it is not a solution to asymmetric warfare. Looking at the larger operational picture, costs in the areas of shipping, fuel to move items, storage, and maintaining items continues to increase globally. Additionally, other payments in the form of infrastructure improvements and “usage” fees to occupied countries are becoming common place for operations that involve even just a modest sized force [133]. The US military can no longer afford to expend costs in areas not intent on directly supporting the force.

The goal of NCW is to outline a framework that can effectively support a variety of military operations via a robust, responsive, and flexible organizational system. NCW proposes the idea of leveraging information age concepts of today and applying them to the strategic environment, thus producing dispersed, semi-autonomous combat capability packages that produce effective mass effects via speed and coordinated efforts [57]. The downsizing of the military coupled with increased requirements has necessitated a fundamental change in how the military does business and sustains itself. Joint Operations require capable organizations that can cross-enterprise and adapt dynamically to uncertainty and turbulence in a multi-dimensional, nonlinear, competitive environment [30]. Under this approach, dynamic decision support systems (DSS) which consists of COA analysis, knowledge management and mining, pattern recognition, learning, and other cognitive decision support tools becomes immensely critical [30]. Other consequences of the joint environment include the capability to perform rapid operations of a smaller yet more lethal force. This requires a flattening of the command hierarchy and a shorter timeframe for intensive conflict operations [30]. Operations will be conducted in non-contiguous space with likely no secure rear areas and without pause. NCW has several key objectives which consist of the following [100]:

- Fight first for information superiority
- Facilitate speed of command
- Promote access to information and shared awareness
- Use dispersed forces for non-contiguous operations
- Promote de-mass-ification

- Eliminate process lines (e.g. move away from serially performing the functions of: organize, deploy, employ, sustain; operations, intelligence, logistics)
- Eliminate structural lines (e.g., move towards joint operations at the small unit level)
- Dynamic and able to conduct self-synchronization
- Have the ability to alter initial conditions

In this environment, the values that aid in the success of NCW are networking, intelligence, environment management, speed or endurance, numbers, risk tolerance, and staying power [100]. In order to accomplish these effects a key enabler identified is the need for a logistics infrastructure and architecture that can operate under the dynamic needs that lie ahead [30], [57], [100], [110].

The DOD, which is responsible for all of the armed forces, is a diverse organization with separate and independently operating organizations. These groups may for one operation interact directly or indirectly to accomplish their mission, while on a different but similar mission interact completely differently. This, for better or for worse, is the nature of the DOD. This type of framework in the engineering world is categorized as a system of systems (SOS). Briefly, a SOS is a system that is made up of several independent systems that have been joined together to perform a specific or variety of functions that leverages the capabilities of each independent systems to result in a better outcome than could have resulted from any of the individual systems alone. This topic will be discussed in detail in Section 2.2.

1.3.1 Modernizing the Current Logistics System

Current efforts to modernize logistics have focused on increasing system efficiency, reducing the mobility footprint in the operational area, implementing performance-based contracts, and creating transparent and global transactional data sharing environments [133]. Technology such as radio frequency identification (RFID) and commercial off the shelf (COTS) tools used in a variety of industries have and continue to be assessed and integrated into the military infrastructure and support systems.

Just-in-time (JIT) logistics, a methodology taken from the business world, was a recent attempt by the military to apply commercial practices to lean-out their inventory practices and make their logistics system more efficient. This concept is born out of the manufacturing arena and attempts to produce and supply resources as close as possible to a user's or customer's forecasted demand, as opposed to producing items earlier with no demand and having to leave an item in inventory or producing it only after an order is received [17]. JIT theoretically allows the implementer to significantly control production and holding costs, reduce inventory, and maximize use of available resources. Its prime metric is *flow time* or *flow rate*. JIT has shown to work well for many applications, but ultimately results in a brittle supply chain that has a high risk of failing in a dynamic environment. This is due to inflexibility, vulnerability to damage and destruction, and potential inability to service dynamically generated or prioritized needs [17]. Although a good start, JIT in its entirety has not solved the need for a dynamic military supply network.

Another application from the commercial sector applied to the military's support architecture is a tool/system called Enterprise Resource Planning (ERP). This system takes a variety of data and processes that exist in an organization and integrates them into a unified system. A typical ERP system will use multiple components of computer software and hardware to achieve this integration. A key ingredient of most ERP systems is the use of a unified database to store data for the various system modules. These modules in the commercial environment consist of the following activities [34].

Table 1: Typical ERP Modules and Activities

Modules	Potential Functions Tracked
Manufacturing	Engineering, Bills of Material, Scheduling, Capacity, Workflow Management, Quality Control, Cost Management, Manufacturing Process, Manufacturing Projects, Manufacturing Flow
Supply Chain Management	Inventory, Order Entry, Purchasing, Product Configuration, Supply Chain Planning, Supplier Scheduling, Inspection of goods, Claim Processing, Commission Calculation
Financials	General Ledger, Cash Management, Accounts Payable, Accounts Receivable, Fixed Assets
Projects	Costing, Billing, Time and Expense, Activity Management
Human Resources	Human Resources, Payroll, Training, Time and Attendance, Benefits
Customer Relationship Management	Sales and Marketing, Commissions, Service, Customer Contact and Call Center support

Besides the above items, ERP also can also contain Data Warehouse and various Self-Service interfaces for customers, suppliers, and employees.

These latest modernizing approaches have produced incremental improvements for the military's logistics support system. However, their inability to provide a holistic solution is revealing that COTS applications will likely not yield the kind of adaptive, effects-based logistics system that is needed to support the highly modular, dynamic, distributed, and adaptive environment the military operates in. In general, military logistics is organized hierarchically and linearly, and does not adapt well to dynamic battle space events. The current logistics structure yields predictive, optimized, and linear supply chains that operate in traditional, hierarchical command and control (C^2) structures [134]. Logistics support in a transformed military and dynamic environment will require [132]:

- Logistics operations in a networked, distributed force that emphasizes speed of command and adaptation

- Focused force-centric logistics for tasks, missions and effects, as opposed to the optimization of logistics supply

- Logistics support for rapid force projection and rapid, decisive operations of expeditionary forces (force to objective maneuver)

Analysis from Operation Iraqi Freedom (OIF) identified four critical areas that will need to be addressed for future logistics support operations. These are connecting the logisticians, modernizing theater distribution, improving force reception, and supply chain integration [134].

To connect the logistician to the information necessary to perform their jobs, the architecture must be capable of connecting critical logistics nodes. These consist of tactical warehouses, ammunition supply points, hospitals and distribution hubs (e.g., airports and seaports of embarkation/debarkation where available or alternatively a Seabase) [134]. The architecture should also provide non-line-of-sight connectivity through enhanced commercial expeditionary data communications via portable satellite communications terminals and integrate Automated Identification Technology (AIT) throughout the joint battle space [100]. Modernizing theatre distribution and supply chain integration consists of effectively inserting *maturing technology* into current distribution platforms, developing intermodal cargo platforms compatible with current and future transportation assets, and improving the strategic distribution process [134]. Improving force reception will involve deliberate and methodical change to all the services to transform their forces incrementally and in coordination with one another to insure the cross visibility envisioned and needed for Joint Operations. The future military logistics structure should be able to forecast logistical requirements to meet the needs of the force.

1.3.2 Sense and Respond Logistics

To meet these needs the concept of Sense and Respond Logistics (S&RL) is proposed under NCW. Merriam Webster's dictionary defines sense as "*a reliable ability to judge and decide with soundness, prudence, and intelligence*". In the context of this definition and application to the military logistics environment, the DM requires the proper information, intelligence, and visibility to arrive at a sound COA for the situation at hand. Once that COA is determined the next step is to respond. Webster's defines response as "*the output of a transducer or detecting device resulting from a given input*". In the context of the current discussion, the military logistician or DM must now formulate decisions that will enable activities and initiate directives that serve to support and sustain active units with the resources required to meet the overall intent of an operation. A methodology that can bridge the gap of determining the most robust logistical architecture between the concepts of Seabasing and DO would be a valuable asset to a military DM.

S&RL is a "transformational network-centric concept that enables joint effects based operations and provides precise, agile support" [30]. It relies upon highly adaptive, self-synchronizing, and dynamic physical and functional processes. The DOD envisions a network able to predict, anticipate, and coordinate actions that provide a competitive advantage over the full range of military operations across the strategic, operational, and tactical levels [30]. Promoting doctrinal and organizational transformation, while supporting scalable coherence of C^2, operations, logistics, intelligence, surveillance, and reconnaissance are also goals. S&RL, it is contended, should be implemented as a cross-service, cross-organizational capability, so that it provides an end-to-end, point-of-effect to source-of-support network of logistics resources and capabilities [30]. Every entity in this network, whether military, government, or commercial, is considered both a potential consumer and provider of logistics. The figure below gives an overview of the connectivity envisioned from the source of support to the point of use in this network.

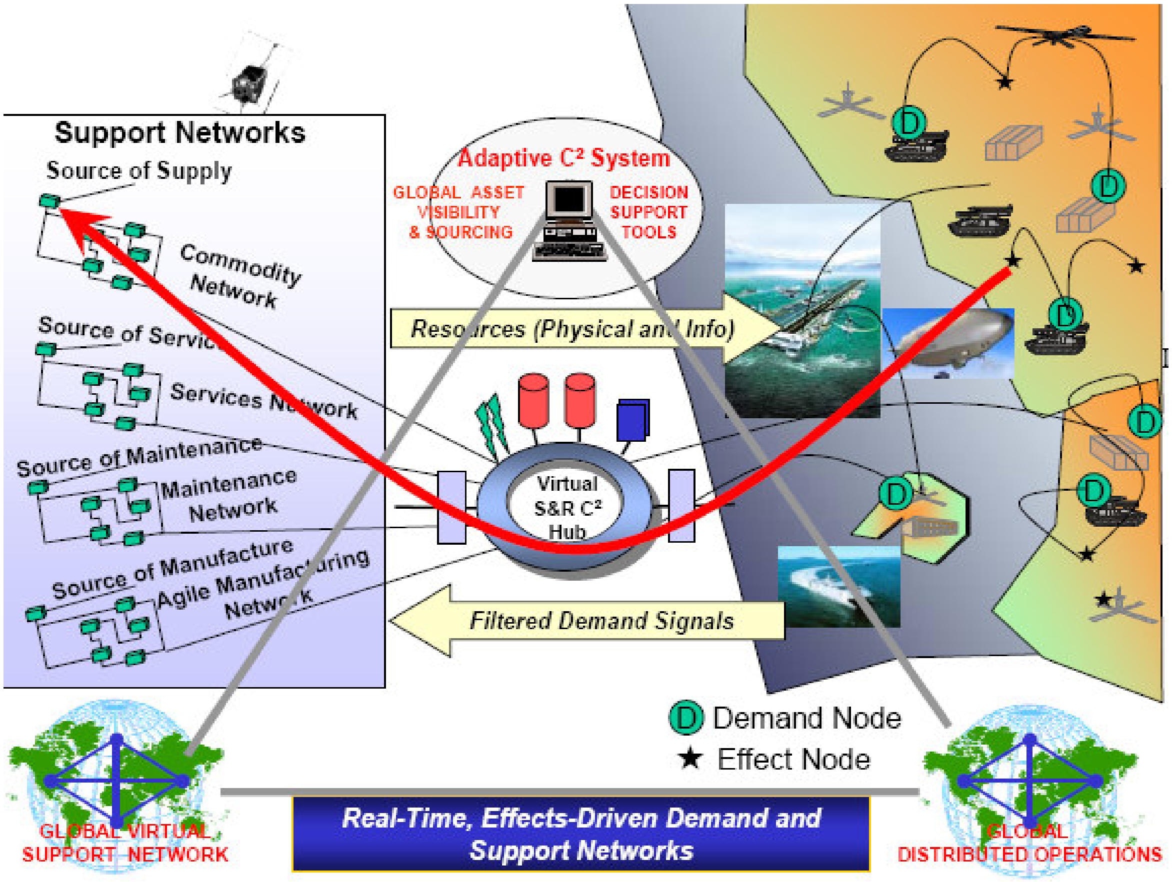


Figure 7: Sense and Respond Logistics Structure Overview [30]

The S&R command and control (C^2) node is the entity that has the logic and architectural framework that processes standards-based transactional information in real time by sending filtered demand signals back to the support networks and constraint information from the support network to the demand network [30]. Things such as lessons learned and experience feedback loops would assist in the adaptation of logistics support strategies over time to meet the dynamically evolving environment. Real-time information flows will link global support networks with total asset and process visibility and global anticipatory demand networks to support effects-based operations and provide a constantly adapting enterprise capability [30]. The concept for operational S&RL under NCW can be seen in [30], but the effects on logistics compared to how it is performed today and the vision for the future are summarized in Table 2.

Table 2: Network Centric Warfare Pre and Post Transformation Comparison [30]

Pre Transformation	Post Transformation
Linear	Non-linear
Chains	Networked
Use-Based	Effects-Based
Service Stovepipes	Cross-Service Mutual Support
Functional Stovepipes	Cross-Enterprise
Title Ten-Driven	Joint Logistics
Pre-Planned	Dynamic Continuous Planning & Execution
Poor Ops/Log ISR Integration	Net Warrior Ethos
Reactive	Anticipator
Parametric Analysis-Based	Collaborative
Hierarchical	Networked
Monolithic	Distributed, Modular
Poor Scalability	Dynamically Scale-able
Not Flexible	Flexible
Consumption-Based	Adaptive, Cognitive
Mass	Speed of Effect
Attrition	Effects-Based
Service Perspective	Joint Coherence
Efficiency	Effectiveness
Highly Optimized	Effective
Brittle, Rigid Supply Chains	Robust, Flexible Demand Networks

Of the many evolutionary requirements to accomplish a S&RL environment under NCW principles, is the need to stimulate innovation, testing, verification and validation of methods and techniques in support of S&RL concepts. The DOD has also identified the "need for experimentation so as to provide a venue for evaluating the S&RL concept and IT prototypes, and provides insights that can be used to refine the concept in an iterative fashion" [30]. From these points it is clear that the military is seeking to facilitate techniques and methods that can support the military decision making process for logistics support activities.

1.4 Chapter Summary and Conclusion

As the mission requirements of the U.S. armed forces increases in scale and breadth, the overall military is seeing its size shrink. This in part due to technological innovations allowing tasks to be done remotely or autonomously, but also because higher costs are increasingly difficult to justify and maintain to policy/decision makers - resulting in pressure to downsize the force. In order for the military to succeed in this environment, its leaders know “requirements for material and supplies must be accurately identified to ensure that stock levels closely approximate needs” [86]. Additionally, military logistics will no longer be able to rely and depend on primarily world class seaports to launch to and from because future operational areas will likely now be in smaller third world countries for reasons given earlier. The areas of operation (AO) will likely have little or no off load facilities thus requiring temporary and flexible logistics infrastructure to provide and sustain material and personnel for the operational environment [86].

The military environment is a dynamic systems architecture and it is one of the most complicated and difficult environments to manage. A dynamic system is one that is time dependent and in a state of constant change. One can make simplifying assumptions to reduce the complexity of the environment, but the assumptions must be valid and reinforced through a logical and knowledge based thought process. With so many moving pieces, in different environments, with different needs, different usage rates, and so on, a clear vision and understanding of the network from end to end is difficult to see.

Time dependant analysis is a difficult task, especially when performing it in an environment with incomplete data or information that arrives too rapidly to digest. In the past, the best one could hope to do when conducting prediction/forecasting activities was to get their forecast right part of the time and hopefully learn from any errors or mistakes. The military can no longer rely on land based support structures and no longer afford to make significant errors if they hope to operate successfully in the new operational environment described. The advancement of technology based capabilities, analysis and other techniques in forecasting betterment should be applied to fill the gap in the military's capability to effectively conduct logistics decision support analysis at any operational level. This belief has spurred the development of a method called Adaptive Technique for Logistics Architecture Selection (ATLAS) that aims to help close this gap and will be discussed in detail in Chapter 4.

1.5 Thesis Organization

This thesis addresses the complexity faced by many decision makers – how to select a course of action (COA) in the face of uncertainty, incomplete knowledge, and more information than is feasibly possible for the human mind to digest. The military logistician faces a complex and nonlinear decision making environment and military logistics is currently the subject of extensive research. Solutions for this environment will require a methodology that utilizes existing information to apply higher level reasoning methods and feedback mechanisms to provide a decision support system (DSS) meant to help solve new problems. DSS's are defined as the combination of the intellectual abilities of humans with the abilities of computer systems in order to improve the quality of the decisions made and to support decision makers in ill structured problems [62]. The aim of this system should be to provide transparent justification for a specific COA and architecture selection based on available information. Before exploring the creation of such a method and system, an exploration of current technology, methods, tools, and software is warranted to become familiar with current state of the art in applications for DSS's. The next chapter of this thesis, Chapter 2, will give an overview of some pertinent concepts that are core to this research and have guided the development of the ATLAS methodology as a DSS to help the military DM with his/her tasks.

Chapter 3 will provide a re-familiarization of the problem and discuss the technical challenges, research questions, and hypotheses for this research effort. Chapter 4 will describe the ATLAS methodology and its steps for implementation in detail. Chapter 5 will describe a proof of concept scenario relevant to today’s military environment in which a DSS can be applied. Also in this chapter a traditional forecasting method that is used by the Army today is assessed and compared to the ATLAS method. This chapter also contains the results of the ATLAS methodology applied with a dynamic case based reasoning tool along with analysis, interpretation of those results, and COA development. Finally, Chapter 6 draws on some conclusions from the work and assesses the hypothesis presented in Chapter 3.

2 LITERATURE SEARCH OF METHODOLOGIES & TOOLS

"As long as our Armed Forces continue to be committed around the globe, our ability to deploy and sustain them will remain a top priority."

-Henry H. Shelton, Former Chairman of the Joint Chiefs of Staff

Today, no one works completely independently when solving a problem or exploring a new design. Almost everyone is part of at least one group, typically several groups at any point in time. Groups communicate, share information, generate ideas, and organize ideas. One of the most important aspects to facilitate determining a solution is communication and coordination, horizontally and vertically, between decision makers and the knowledge they need to make effective decisions [20]. To accomplish this need, decision support frameworks are usually generated and made up of several components to draw upon. Each system will differ depending on the need, but there usually exists a knowledge source or database of relevant and pertinent information to draw upon. These databases can have either quantitative or qualitative data, but often times have a combination of both. Qualitative data usually exists in the form of expert knowledge and tends to be difficult to apply numeric values for a variety of reasons. This data is usually represented with some form of industry specific language or linguistics. Quantitative data usually consists of hard factual data that has been collected, measured, or viewed with discrete or continuous inputs and outputs recorded. These types of databases, especially the hybrid consisting of both data types, are becoming more and more prevalent due to technological advances in the areas of artificial intelligence, and computer processing that attempts to simulate the human though process of solving problems. Along with a database there usually exists underlying software that is used to model the environment and is able to visually and synthetically represent the investigated design space. The use of surrogates or a model of this model can be created and used for quick and limited design space exploration by a user. This is usually accomplished via a user interface node that allows for the processing of new information or the correction of information already stored in the database. All of these components make up a general architecture that is often referred to as a decision support system (DSS). This is represented in Figure 8 below. Critical to any DSS is knowledge management.

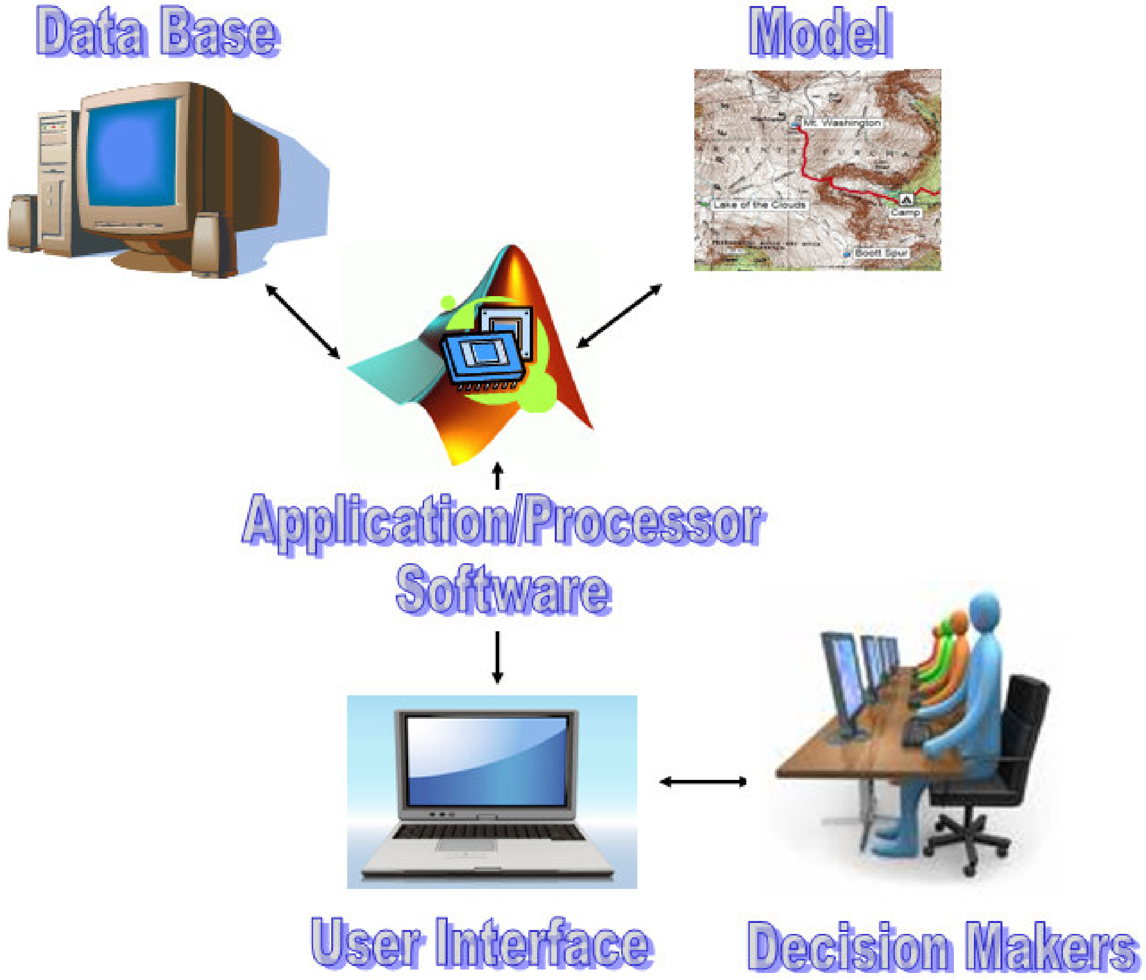


Figure 8: Decision Support System Basic Architecture

The science of Knowledge Engineering has evolved to encompass several methods and techniques to aid decision support architecture generation. Many problems today are characterized by abrupt changes and increasing uncertainty resulting from a variety of noise variables that are very hard to characterize or understand, but are represented linguistically. Formulating a model and ensuing scenarios that capture the critical aspects of the problem is more of an art than a science and requires experts from the field to be heavily involved in the development of these systems. Critical to developing a useful DSS is an understanding of the problem and any intricacies within the design environment. Doing this lays the foundation for an effective process, methodology, or tool to ultimately be developed. For the military, the problem they face is a supply vs. demand problem, i.e. the goal of getting the right logistics to the right people and at the right time. The author investigated existing and relatively new techniques to see if they were viable and suitable to be applied to this dynamic environment. Complimentary to this literature search, was a search for applicable tools incorporating state of the art techniques in forecasting and predictive analysis, especially in the area of military DSS.

2.1 The Importance of Forecasting: Supply vs. Demand

Matching supply with demand, that statement seems undaunting, but in reality it is an extremely difficult task to accomplish when operating on a global scale as the US military does. In economics there exists the theory that price is dictated where a theoretical supply and demand "curve" or line intersect. If there is excess demand and supply remains the same, prices will rise. If there is excess supply and demand remains the same, prices will fall. This is shown in Figure 9.

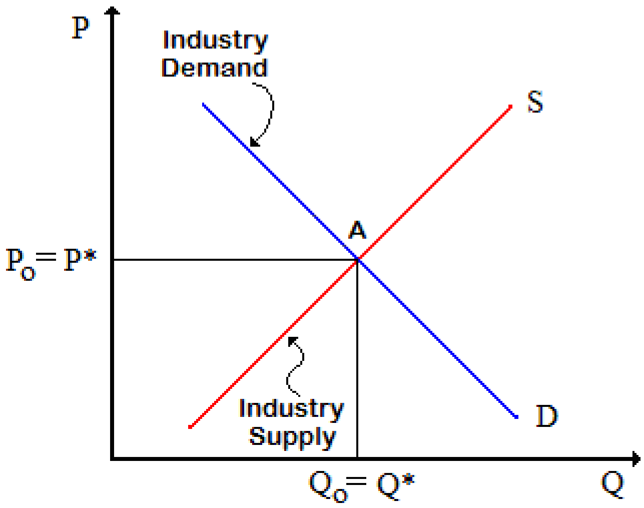


Figure 9: Supply and Demand in Equilibrium

Not only are market forces in effect, but uncertainty in the environment for which the product is in and uncertainty of extenuating circumstances exist that rarely allow supply and demand to ever reach equilibrium, point A. In any field where the supply and demand quandary is faced, it is known that no one will actually attain a stable equilibrium. This is accepted primarily because once equilibrium is "reached" other market forces will fluctuate and cause a change in the supply and or demand curve. In Figure 9 two "curves" representing supply and demand, say of automobile gas, intersect at point A where the current price is P_o and the current quantity demanded is Q_o. As time progresses by some small increment other factors generally cause a shift in the supply (S) or demand (D) curve and sometimes both. In Figure 10 an increase in demand causes the demand "line" to shift up and to the right represented by D'.

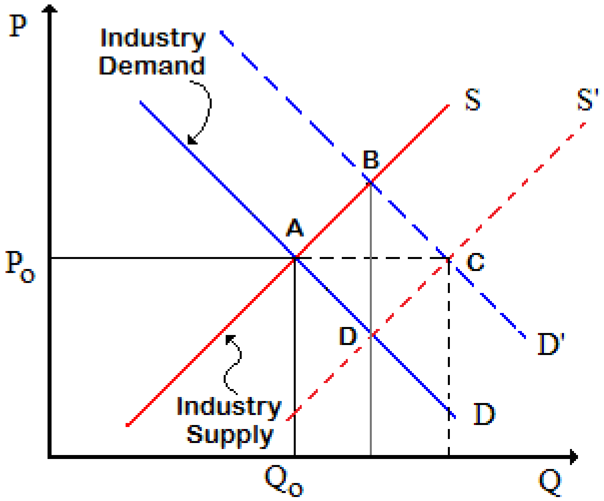


Figure 10: Supply and Demand Curve Shifts Due to Changing Market Forces

If supply stays the same the effect is an increase in price to point B, and gas gets more expensive. If more gas is produced to meet higher demand, the supply line moves to S’ and we see price fall back to point C which is equivalent to point A. The initial effect of increased demand allowed those providing oil for gas to enjoy a premium price on their goods, but not for long since an increase in supply through increased production, caused the price to come back down because the market experienced excess available product. In mid to late 2008 consumers saw OPEC increase the supply of oil for a short time because of world pressure due to significantly increased gas prices. Initially demand artificially increased due to natural weather phenomenon, primarily in the US, that produced a gas shortage fear by consumers. As a result consumers purchased gas they ordinarily would not have bought. This caused an artificial spike in demand in gas and a perceived shortage. Prices increased dramatically in a short amount of time. When supply increased slightly and demand lost its artificial increase by settling back to the original line, D, gas prices fell and have even fallen lower to prices reminiscent of 2003 prices. This cycle will continue with a reduction in production that should increase price back up to an "acceptable" market price where suppliers and consumer reach a general agreement on what each is willing to buy and sell a gallon/liter of gas at. Why is understanding this concept important?

Understanding how "market forces" and other environmental variables have an effect on the supply and demand environment, such as the military's logistics network, can be beneficial in trying to forecast needs based on analysis and observation of certain environmental variables. While economist might find some solace in the supply versus demand theory and will likely continue to contemplate its philosophical context, those who are responsible for delivering and managing commodities to consumers desire a clearer picture of the operational environment to aid in their decision making activities. For these DM, excess demand can mean increased costs, for example in the form of paying a premium for increased productivity for a resource that is not in short supply or needed right away. On the other hand, excess supply can mean wasted resources, say in the form of too much inventory of ordnance at one location while it is desperately needed somewhere else. It can also mean excess costs in the form of holding costs because of too much inventory. On average, most organizations generally find themselves with the correct amount of resources (people, products, and or equipment), but find themselves with the incorrect amount of resources in the wrong place, at the wrong time, and/or in the wrong quantity [17].

The airline industry, for example, utilizes load factor to attempt to at least break even by having revenue meet operating cost. Load factor is determined by a several variables and is constantly changing depending on which model is being used. In general, for every increment above the minimum load factor calculated by an airline they will gain an increase in revenue and make a profit. Consider an airline, whose percent utilization of an aircraft is at 70%. That means that with a 300 seat aircraft 90 seats do not need to be occupied to break even. If the airline were to be able to have one more seat occupied per flight, thus reducing the number of open seats to 89 on an annual basis, they would see and increase in utilization by approximately $0.3\underline{3}\%$ per flight performed. The net effect of this change, if one looked at the 2001 balance sheet for British Airways would equate to a $65 million annual increase in profits for their 300 seat aircraft [17]. One seat makes that much of a difference. Considering the number of flights an airline conducts over a year and the amount of profits one additional customer can provide, this should start to show the reader how important and critical forecasting can be when the outcome is based on a large scale or volume of resources.

How does all this relate to logistics the reader might ask? In the context above, people are the commodity and the airline is the user trying to predict the logistical need (number of seats on airplane) to break even and or make a profit. The U.S. military is a large consumer of resources and commodities during peace time, but especially so when involved in any type of operation that requires above and beyond planned supply levels and delivery. The military too is faced with the conundrum of getting resources, people, and/or equipment to the right place, at the right time, and at the correct quantity to achieve success in its missions. The need to be able to accurately forecast and predict these needs is of strong importance and concern across the defense community. In the first draft of the Joint Integrating Concepts document put out by the Joint Chiefs of Staff in early 2006 the following, of many recommendations, in the arena of improved logistics was set forth in the area of predictive analysis.

> There is the need to focus and "conduct experimentation to further develop predictive analysis of sustainment requirements. Recommend emphasis on application of initial sense and respond logistics areas by the joint deployment and distribution enterprise (JDDE) in support of the joint force commanders (JFC)" [58].

Other areas of emphasis also related to logistics concerned the determination of the optimum number of lift assets needed to support future operations and a look at multi echelon priority systems, of which the goal is to identify supply priorities in a theatre where there exist competition for scarce commodities [58].

The task outlined above is extremely complex and large with many dynamic and interconnected pieces; not something easily solved in a short amount of time or by one person. What is feasibly solvable is a slice of this environment in the area of Seabasing and Distributed Operations. Seabasing and DO were briefly described for the reader in Section 1.1.3 through Section 1.1.5, but even this part of the future sense and respond environment for the military is too large. Finding a portion of Seabasing and DO that is of manageable scope and size to apply a methodology in predictive techniques and analysis is the goal. Before potential new methods for forecasting a dynamic environment such as the concepts of Seabasing and DO can be investigated, a systematic research effort needs to be performed to see if there existed any viable techniques capable of modeling, analyzing, and simulating the proposed micro-environment and its variety of subsystems.

2.2 System of Systems (SOS)

This field of engineering was born out of the realization by many in academia and industry that one can no longer work in isolation on a problem or design. There exist too many factors, complexities, and interrelated fields for one to think that all that matters in the endeavor is one specific field. While some disciplines may be more important than others, collaboration and cooperation are becoming more intensely necessary in much of what is done today when looking for solutions. As knowledge and technology progress at a faster rate than it did the year before, working together has become an essential ingredient to success in many endeavors. Long before the phrase system of systems was utilized to describe the collection of independent systems working together to offer a more robust environment, there existed systems engineering.

As a systems engineer, a person was responsible for looking at a project, problem, or design from a holistic point of view. It was his/her job to have an unbiased view and be able to see the needs of each discipline involved, but to also be able to integrate them properly in such a way that a feasible product or design was the end result. Systems engineering arose because the things we do as a world started to get more and more complex. Design of an aircraft wing, for example, in the early stages of aviation iterated on a particular shape or wing characteristics that was known to work. The tools and methodologies to do in depth design and analysis were limited or the calculations were insurmountable. As the computer and software technology developed this began to change. Increased complexity in aircraft design arose because of many factors, some of which were more powerful engines, transitioning from propeller to jet engines, change in flow properties, and assumptions around the aircraft that were no longer valid. All of these changes necessitated a new way of looking at the processes followed for design solutions.

This new way of looking at processes or how tasks are approached called for a change in thinking or a paradigm shift. Mavris et all in [80] discuss this type of change as a result of the focus on design for affordability and the desire to look past just the development and implementation of a design, but to also look at the product life cycle from "birth to death" of a product. This new focus and an attempt to balance mission capability with other system effectiveness attributes, while keeping cost under close attention was dubbed *Design for Affordability* [80]. The three metrics focused on were incurred cost, knowledge about the problem, and freedom available to the designer. Figure 11 is inspired by this premise, but applied to a DM's environment. It shows initial costs tend to be high with knowledge uncertainty and costs diminishing as knowledge gathering is conducted, represented by the orange line.

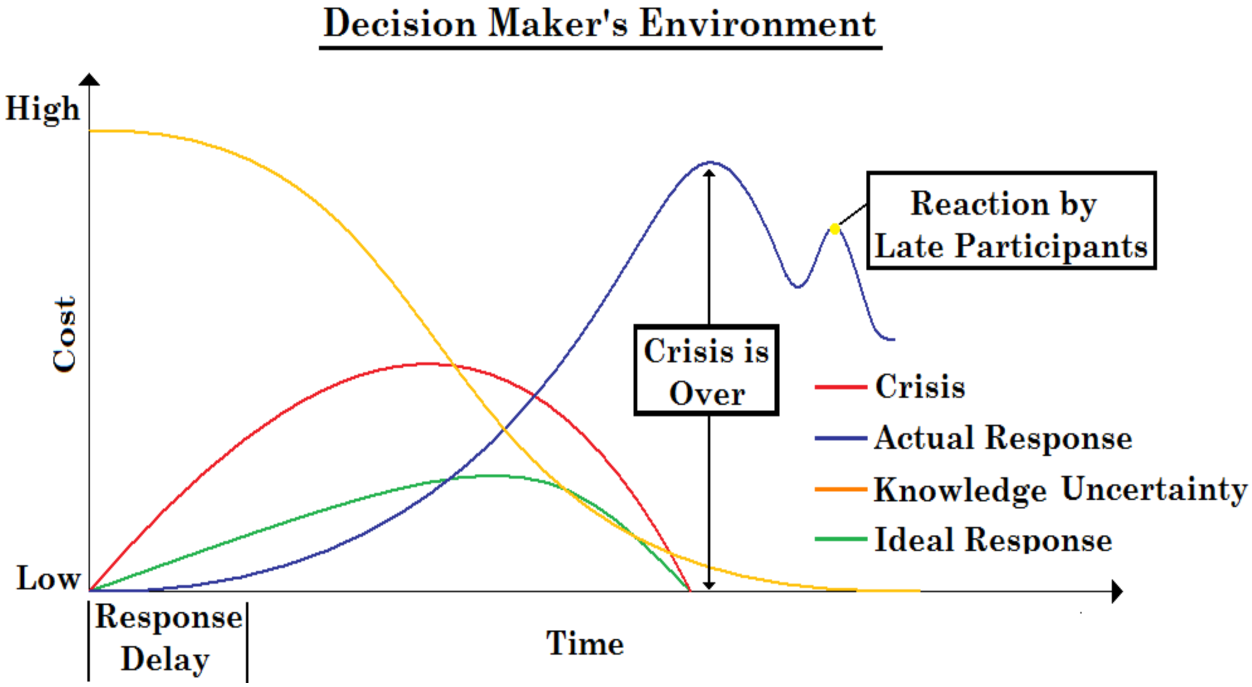


Figure 11: Environmental Depiction facing Decision Makers

The goal for any DM is to gather information quickly that is pertinent and useful to determining a COA to an event, represented by the red line. The typical reaction by

DM's is represented by the blue line and the goal is to transform this response so as to reduce cost and reaction time according to the green line in Figure 11. What Mavris et al proposed was the need to bring in physics based codes, enact concurrent engineering, and integrated product development earlier into the design process possibly as early as research and development; ultimately bring more information into the early stages of the process [80]. This approach can be applied to other areas where finding a solution to a design or problem is concerned. Design for affordability and the consideration of life cycle costs are the norm, not the exception today. As technology changes at a rapid pace and new capabilities emerge, it is the designer or engineers that will need to implement them so that cost can be minimized and quicker reaction times can be realized.

With this in mind, no longer is it the sole goal of companies to design something new as the result of a successful past product. Now systems engineers and designers are faced with the need for what sometimes requires a robust revolutionary concept, design, or product to meet future needs. Not only is the engineering environment changing, but the customer or end user environment continues to change. Incorporating customer requirements and wants is now a factor and adds to the overall complexity of the solution process. There are many factors that increase the complexity and this list tends to grow as today's problems are tackled. How systems engineering interacts with all facets of a process, whatever that process is, can be seen represented in Figure 12.

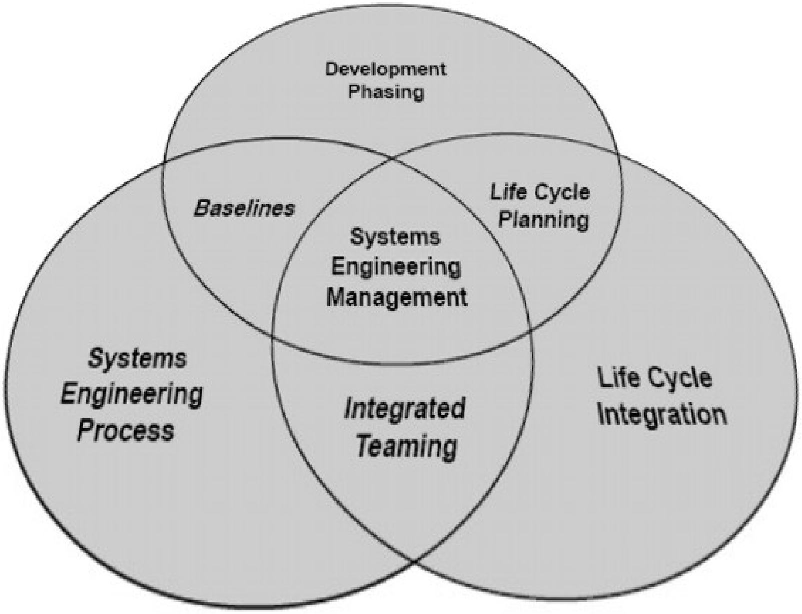


Figure 12: The Scope of Systems Engineering [125]

There is no set process when it comes to systems engineering, but there are a few general guide lines that many tend to follow to insure a holistic approach and to gather as much information early on as possible. The first step is generally to define the problem through an evaluation of requirements or needs for a design or project to fulfill. Truly understanding these requirements is critical. Failure to perform this step can potentially cause failure to the effort before it ever really begins Examining what already exist that may fit the needs or what is say a 90% solution, but is lacking in capabilities may be next. Keeping an open mind and looking to other disciplines that have been innovative in their approach to solving a problem is another approach. This functional analysis should help all involved understand the problem even more and enable them to become intimately familiar with critical aspects of the design or problem. This general process is depicted in Figure 13.

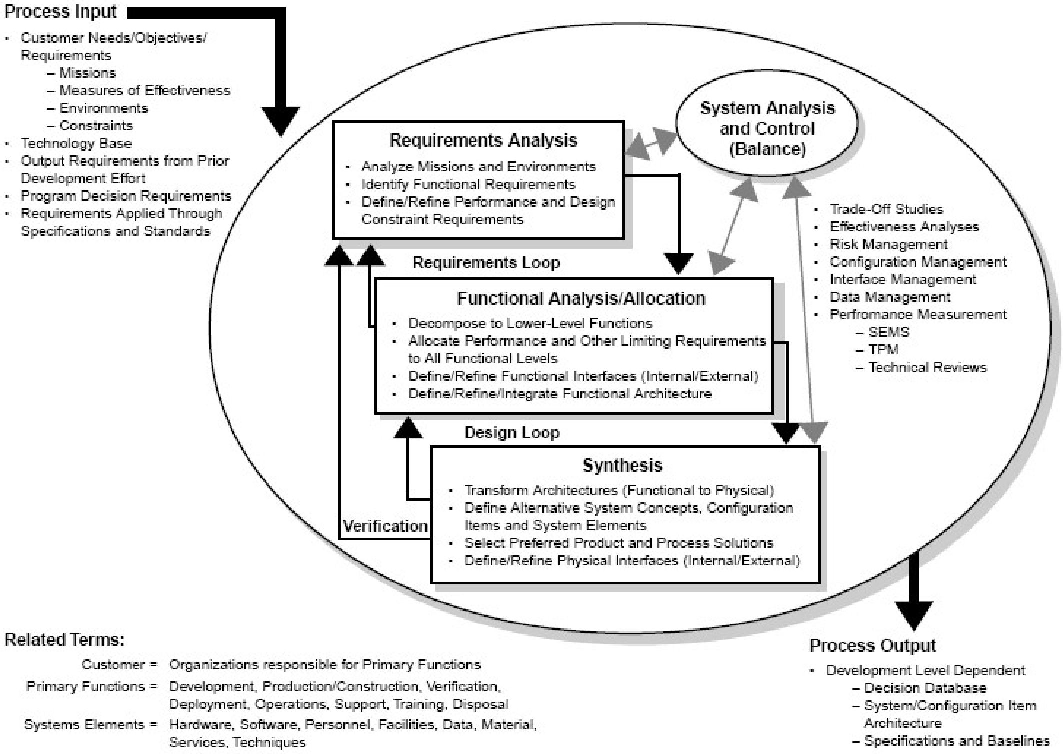


Figure 13: General Systems Engineering Process [126]

Once this has been accomplished design commences and it is the systems engineer’s primary responsibility to guide the project in a direction that promotes a successful design or resolution. Out of this environment is born the system of systems approach. This has been both an evolutionary and revolutionary process. There is no absolute definition of what constitutes a system of systems, but many tend to agree that they typically exhibit the behaviors of complex systems in which their interactions typically expose and deliver important emergent properties [124]. Five characteristics proposed and commonly known as Maier’s criteria attempt to categorize the identification of a SOS environment. They are managerial independence, operational independence, collaboration amongst the sub-systems, emergent behavior, and evolutionary behavior [73]. These five have been further augmented by also adding the traits of trans-domain, heterogeneity, and a system of networks proposed by De Lauerentis [28].

The first five criteria are essential in identifying a SOS if not merely for the fact to help other engineers identify what a SOS environment is, but also because as things get more and more complicated and intertwined, a designer/engineer could easily begin to classify everything as a SOS without some form of generally agreed upon guidelines. The definition of managerial independence is that the components of the systems are separate and independent systems that are separately acquired and integrated, but continue to maintain and independent existence of the SOS [73]. The definition of operation independence is that if the SOS were to be separated into its separated systems, they would be able to operate usefully and fully on their own by fulfilling some oriented purpose. Collaboration is the premise that these subsystems are not only working together to solve a complex problem, but they are also doing so freely. The thought is that the separate system for whatever reason, conscious thought or a decision to integrate is more beneficial than trying to operate independently. This does not necessarily mean that the SOS is the most efficient way of doing something. On the contrary, it can be more expensive especially if there is not one process owner or member that can see the benefit of managing the SOS to that end. Emergent and evolutionary behaviors consist of new and generally unintended consequences of the grouping of collaborating systems. An example of this is the World Wide Web. Its intended use was to exchange scientific data between government scientist and it has transformed into a medium that has given the world access to one another in many ways, shapes, and forms.

The final three derivative criteria, introduced by De Lauerentis, attempt to further define the SOS environment with the following definitions [28]:

- Trans-domain - (Proposition) Effective study of SOS requires unifying knowledge across fields of study: engineering ∪ economy ∪ policy ∪ operations.

- Heterogeneity - Constituent systems are of significantly different nature, with different elementary dynamics that operate on different time scales.

- Networks - Define the connectivity between independent systems in the SOS through rules of interaction.

These traits further give the reader an idea of what types of problems can be considered as a system of systems. Using these defining characteristic should aid in the process of moving forward to categorize problems, define core architecture characteristics, and ultimately find a design point or family of solutions to a properly laid out and defined problem.

In a SOS framework, resources, operations, policies, and economics are used to categorize roles of the involved elements and their interactions [120]. The complexity associated with today's SOS environment is the subject of intense focus for many of the reasons that necessitated the field of systems engineering in the 1940's. This reason chiefly is increased complexity of the problem being analyzed. This field of research continues to grow today as researchers, academia, and industry continually evolve and attempt to solve problems that in the past were considered impossible because of constraints that are being resolved today at a quickening pace and the ability to manage and process information/knowledge quickly and efficiently.

2.3 Knowledge Engineering

Knowledge Engineering (KE) is a discipline that attempts to codify information that can be quantitative, qualitative, or a combination of the two into a suitable format for computer application and processing [113]. One of the most recent techniques developed to aid in this transformation is artificial neural networks (ANN). The concept attempts to mirror the basic functions the most complicated and efficient "computer" known to man; the human brain. To this end, a set of basic principles have been established to help a designer create a knowledge-based system. KE involves two overarching tenants which are *Knowledge Acquisition* and *Knowledge Codification* [113]. Both of these actions are interdependent of one another because the designer must determine whether the information or data being collected can be codified, but also must be able to some how acquire relevant and pertinent data. The reader is directed to [81] and [112] for further information on the topic of KE.

The two ways that knowledge is garnered in just about any situation generally stems from operational data and or user experience. Operational data in a military sense usually comes from executing missions under varying initial conditions and recording the outcomes. This information, unfortunately, is very hard to come because it is classified, convoluted, or data was never effectively recorded. Overcoming these hurdles can often be accomplished through modeling and simulation of the problem environment. Building these models can range in complexity and it is important for the designer or researcher to thoroughly understand the problem so that a representation created to conduct design space exploration and analysis can at least be verified and validated for use in the solution process. There exist several concepts or techniques under KE that are used to promote the advancement in hybrid quantitative and qualitative environments. They will be discussed next.

2.3.1 Soft Computing

Lofti Zadeh, a pioneer in developing computing methods, makes an observation in [143] that the machine intelligence quotient has markedly increased over the years and that there are many factors responsible for this, but in his opinion the most important factor could be referred to as soft computing, more explicitly *fuzzy logic*. Prior to exponential advances in the computer, mimicking the abilities of the mind was very difficult and computing primarily consisted of precision, certainty, and rigor [143]. Fuzzy logic allowed a departure from this by allowing for exploitation of imprecision and uncertainty. This concept attempts to overcome the curse of dimensionality that in effect states that as the number of features, variables, or dimensions of a problem increase, so does the amount of information and capability necessary to process it. The ability to allow for imprecision and uncertainty by humans is apparent in every day life. We use linguistic to describe how we feel or level of pain to medical practitioner. We use words to describe time duration like long, short, forever, etc. It is up to us, as processors to determine what those words mean.

“Soft computing uses the human mind as a role model and at the same time aims at a formalization of the cognitive processes humans employ so effectively in the performance of daily tasks” [143]. Soft computing tenants are applied more and more today and have proven to be very effective at modeling real world events and representing environment where human in the loop (HITL) interaction and decision making is unavoidable. The concept of fuzzy logic will be discussed in Section 2.10.

2.3.2 Analogical Reasoning (AR)

Analogical Reasoning (AR) research focuses on basic cognitive mechanism such as matching, retrieval, reasoning, and learning [26]. The goal of AR is to mimic the ability of the human mind to logically and methodically go through the decision process to come to a suitable answer, one based on the data presented. This is usually accomplished by applying statistics in a variety of ways to help solve problems. Several formalisms that are capable of modeling some ones degree of belief are Bayesian Theory, Dempster-Schaeffer Theory, Information Gap Theory, etc. Dempster-Schaeffer theory, sometimes called the Theory of Evidence, does not need prior knowledge of a probability distribution because it assigns probability to sets of possibilities rather than a single event. One of the benefits of this theory is that it is capable of modeling more precisely based on natural reasoning that is conducted by human beings when they gather information to come to a decision or course of action to take [19]. Success in these applications has shown promise for generally simplistic models, unfortunately modeling the complexity of full human cognition has proved to be very elusive. For that reason a great amount of research continues in searching for the best way to represent this ability synthetically via a computer.

Info Gap Theory was developed in an attempt to overcome limitations inherent in probability theory. It also seeks to solve problems with limited amounts of information. A model using Info Gap Theory groups sets of values of uncertainty into one potential value of uncertainty based on each sets level of uncertainty. To accomplish this task, the robustness function and the opportunity function, are used. Ben-Haim in [11] describes the robustness function as the largest level of uncertainty that does not violate a constraint and the opportunity function is defined as the lowest level of uncertainty that allows for the possibility of success. Refer to [11], [19], and [26] for more information on AR techniques.

2.3.3 Case Based Reasoning (CBR)

Case Based Reasoning (CBR) is concept that is inspired by the explosive research in artificial intelligence applications in computing. Its foundations lie on a set of well established scientific fields consisting of but not limited to cognitive sciences, knowledge representation and processing, machine learning, and mathematics [112]. CBR is a concept that attempts to organize information in a logical and useful manner by utilizing a varying combination of these disciplines to represent holistic information via a database that can be drawn upon to reduce decision time and uncertainty in the decision making process. The purpose of querying a database of these stored cases differs depending on the application, but they are generally for two purposes [1]:

- To try and predict or forecast the future environment based on initial conditions or similar environments
- To identify correlation or emergent behavior between initial conditions and outcomes not readily apparent from single events

Technology advancements have allowed people today to be exposed to an increasing amount of information and data. Although humans are very adept at problem solving in most cases, they do have their limits in so much as how much information they can process. Computers on the other hand are virtually limitless in memory capacity and are continually getting faster and faster in terms of processing power, storage, and computation speed. CBR can be traced back to the late 1970's and the general model can be described as a cyclical process that is made up of the *four REs* (Retrieve, Reuse, Revise, and Retain) and is displayed in Figure 7 [1]. The author has added a new step that could occur if solution is not found. This step is *Redefine* and would take the place of *Revise* and *Retain* in the traditional CBR process if iteration of the process is needed.

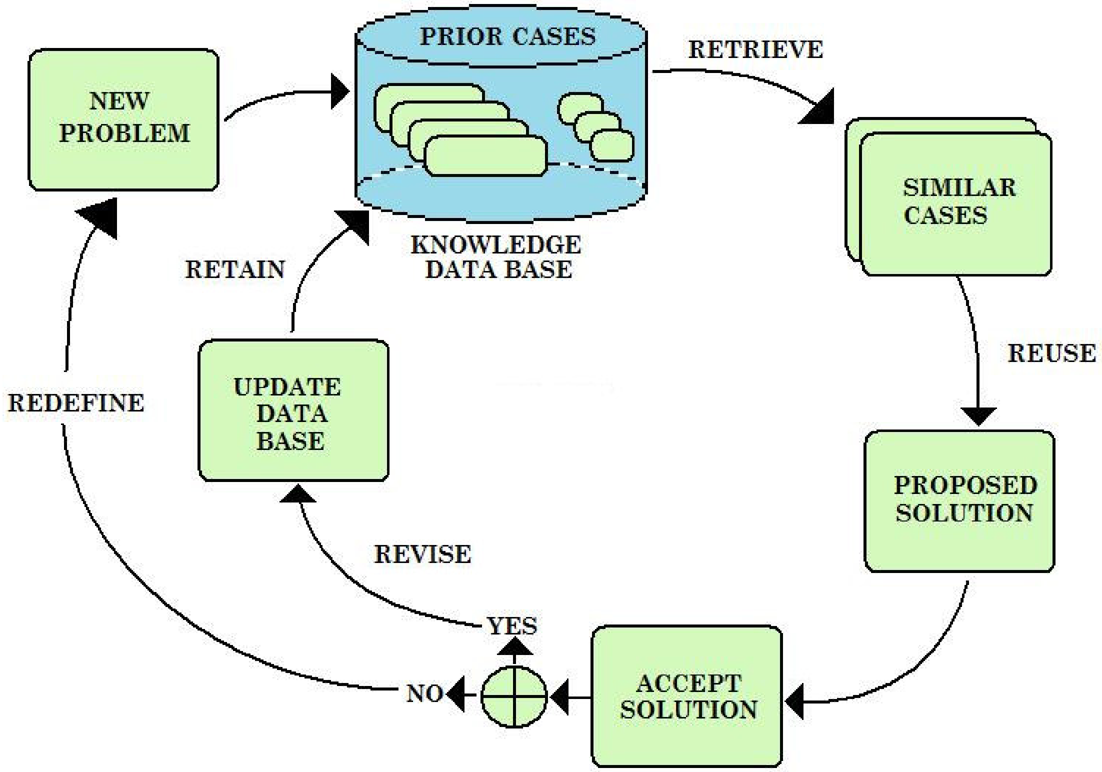


Figure 14: The CBR Cycle, Adapted from [26]

CBR is founded in an observation that successful problem solving is rooted in applying similar solutions to the problem at hand. Learning is based on the concept that a solution was successful or partially successful. The use of AI along with quantitative information is prevalent in many facets of today's operations. An example of this is prognostic health management systems in aircraft engines. This approach and technology helps to identify problems well before they can become catastrophic. Figure 15 shows how sensors are incorporated to relay information to a DM to assess aircraft engine health.

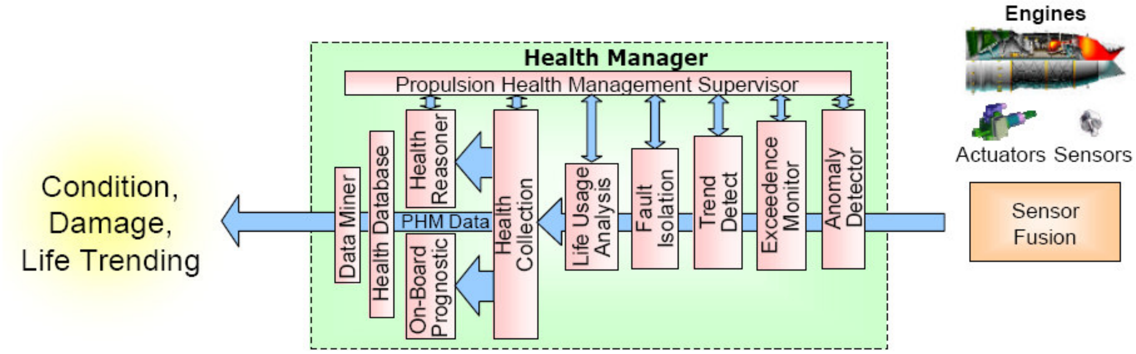


Figure 15: Propulsion System Health Management System [85]

CBR and Artificial Intelligence (AI) are two complimentary concepts that have been the source of great research in recent years as outlined in [26], [60], [87], [112], [121], [123], [140], and [143].

Solving a problem with CBR first requires evaluating a problem and measuring the similarity of the current input characteristics to the input characteristics of a previous problem along with its associated known outcomes or proven solutions. This technique is referred to as *k Nearest Neighbor (kNN)* and seeks to find the most similar case or cases to the problem currently being investigated. Often times this accomplished by calculating the Euclidean distance between a case vector and query vector. There are several methods and techniques that apply weightings schemes and incorporate a variety of other methods like neural networks to facilitate this process. Next in the process is to retrieve one or more similar cases and applying the solution process proposed by the system, generally with some adaptation to account for any differences. These adaptations are usually proposed and applied by experts who are HITL participants. The resulting solution is then retained in the knowledge database as a “new” case that can be appended to the information and solutions the database has available. Sometimes reuse of a case based solution presented during a CBR query results in a solution that is significantly different between the new case and the retrieved one. In these circumstances adaptation or revision of the information may be adopted to account for these differences.

Adaptation becomes ever necessary when constructive problem solving tasks such as design, configuration, and planning activities are being executed [26]. For tasks such as these the likelihood that an exact case or very close case is relevant or present is low, so the solution is an initial solution and will likely need some iteration or analysis performed upon it. In applications such as classification and diagnosis, an attractive feature of CBR is the ability to explain the predicted outcome by showing the user one or more of the target problem's nearest neighbors. As noted by Leake "the results of CBR systems are based on actual prior cases that can be presented to the user to provide compelling support for the system's conclusions" [67]. However, in the literature search several authors have questioned the effectiveness of precedent based explanations in which the user is simply shown the case that is most similar to the target problem. Much research in the area of what is the best methodology to apply CBR to a problem is underway as is evidenced in [26]. This reference gives an excellent overview of the state of the art and applications many researchers are investigating and conducting in the area of knowledge management through CBR. For further applications of CBR the reader is referred additionally to [1], [60], [67], [90], [91], [94], [104], [112], [113], and [123].

2.4 Multi-Attribute Decision Making (MADM) Methods

Decision makers throughout history have continually developed methodologies to assess, evaluate, and select the most suitable course of action given a set of goals or objectives. The advent of technology, especially in computing has allowed much of the intensive mathematical calculations that once prevented quick analysis and results from occurring to now become a reality. These methods are often referred to as Multi-Attribute Decision Making (MADM) methods. As with any decision making process there is always more than one solution and often times the ranking of the best solution is driven by the user's preferences.

2.4.1 Distance Metrics

A distance metric, *d(x,y)*, is used to measure the similarity of two vectors, x and y, in a space of N dimensions. The distance metric must have the following properties [108]:

1. The distance between two identical vectors is zero: *d(x, x)* = 0.
2. The distance between *x* and *y* is the same as the distance between *y* and *x*: *d(x, y) = d(y, x).*
3. The sum of two distances must be less than or equal to the sum of the individual distances: *d(x, z) ≤ d(x, y) + d(y, z).*

The Euclidean distance between two *n* dimensional vectors X and Y is defined as:

$$\mathrm{D(x, y)} = \sqrt{\left((x_1 - y_1)^2 + (x_2 - y_2)^2 + \ldots + (x_n - y_n)^2\right)} = \sqrt{\sum_1^n (X_i - Y_i)^2}\,, \quad \text{Equation 1}$$

where, X = $\left(x_1, x_{2,}, x_3 \ldots x_n\right)$ and Y =$\left(y_1, y_{2,}, y_3 \ldots y_n\right)$. The sum squared difference metric is sometimes used in place of the Euclidean distance to save processing time. For the same two vectors previously define, X and Y, and of length N is defined as:

$$d(x, y) = \sum_1^N (X_i - Y_i)^2 \qquad \text{Equation 2}$$

Distance metrics help to determine how close a feature or query vector is to another. If the query vector is with in a specific distance to the desired vector, then the pattern can be classified as a member of the desired vectors group. These metric can also be used to determine how suitable a course of action vector is to past event vectors. In terms of meta-models that use artificial neural networks, these metrics are widely used in various forms of supervised and unsupervised training and execution of the neural network. Neural Networks will be discussed further is Section 2.8.

2.4.2 Technique for Ordered Preference by Similarity (TOPSIS)

A popular technique for ranking alternatives is a method called the Technique for Ordered Preference by Similarity to the Ideal Solution (TOPSIS). TOPSIS performs the ranking by normalizing all of the metrics that are used to measure the potential solutions ability to meet the required objective [50]. TOPSIS facilitates the creating of a positive ideal solution that has the best attributes from each of the potential alternatives and also creates a negative ideal solution that contains the worst attributes of each potential alternative. Each metric is then normalized and weighted based on a relative importance of the objective or goal being sought. Utilizing the Euclidean distance, discussed in the previous section, each potential alternatives distance from the positive and negative ideal is calculated and then the alternatives are ranked based on the distances from closest to the positive ideal or furthest away from the negative ideal solution.

2.4.3 Overall Evaluation Criterion (OEC)

The Overall Evaluation Criterion (OEC) allows decision makers to get a single measure of "goodness" of the *N* number of alternatives being investigated. A baseline alternative, $\text{Obj}_{1..N,\ Baseline}$, must first be established to measure alternative against for OEC to be performed. Once this is done, the alternatives, $\text{Obj}_{1..N,\ Alt}$, can be measured relative to the base line's characteristics. The following equation is displays how the OEC is calculated when the goal is to maximize an objective, like the number of enemy combatants eliminated:

$$OEC_{Alt,x} = \beta_1 \cdot \frac{Obj_{1,Alt.x}}{Obj_{1,Baseline}} + \beta_2 \cdot \frac{Obj_{2,Alt.x}}{Obj_{2,Baseline}} + ...\beta_N \cdot \frac{Obj_{N,Alt.x}}{Obj_{N,Baseline}} \qquad \text{Equation 3}$$

where, the β_i term represents the weight, generally between zero and one, specified by the decision maker for each particular criterion. This allows a decision maker to allow select criteria in the overall problem to carry more or less weighting if the criteria is deemed more or less important than another; otherwise, $\beta_{1..N}$ can be evenly distributed by dividing one by N if all criteria are to be treated with equal weight. If the goal is to minimize the objective, i.e. to minimize friendly casualties, the *Obj* terms of the above equation would need to be inversed. This technique is a simple, but very effective way to compare various alternatives to a baseline solution.

2.5 Design of Experiments (DOE)

A Design of Experiments (DOE) is an approach to solving problems through data collection that enable valid, defensible, and supportable conclusions [96]. These experiments attempt to explore a design space via a set of ranges placed on input variables that are thought to uniquely and accurately define the output or responses of a problem being analyzed. The inputs to the DOE are orthogonal to insure that the effects of each variable are not correlated with one another [96]. Computer models and simulations coupled with a DOE allow for the automated execution of a wide range of settings to explore and investigate potential outcomes in a design space. DOE's have been developed to provide insight to different areas of the design space. A full factorial is the most exhaustive experiment that can be conducted as it explores every possible option. The number of experiments for a full factorial is determined by the equation i^n, where n is the number of variables being investigated and i represents the number of discrete settings. For an experiment that has 14 variables and two settings, a full factorial DOE results in 16,384 cases to be executed. Other DOE's have been developed such as the Central Composite and Box-Behnken designs that significantly reduce the number of simulations that need to be run and still retain a high level of fidelity when exploring the design space [16].

2.6 Scenario Approach

These types of approaches were developed as a way to aid in DSS's for strategic analysis and planning. They consist of carefully examining and constructing a model that represents a set of potential future states. The goal is to provide synthetic information by representing a complex network of the problem being analyzed. The scenario is supposed to give decision makers a better understanding of what futures could lay ahead. Scenarios are extremely popular in today's world of modeling and simulation especially among military planners and analyst and corporate decision makers [88].

Scenarios carry several definitions depending on what context they are being utilized for and the goal of the user. Some feel scenarios are cognitive maps that allow for the quick update and search of relevant information. Others view them as potential future states that aid in clarifying alternatives, while some use them to try and forecast or predict the future. Scenarios can generally be classified into descriptive and normative. Descriptive scenarios are characterized as providing insights into cause and effects relationships, where as normative scenarios explain how to achieve a set goal [76]. These types of approaches have always been a part of the military environment [88]. Trying to predict what an enemy might do during battle or conducting analysis of potential hotspots around the world helps the US military to be prepared for action, immediate and long term, as well as aids in formulating direction for training, research and development of new technologies, and decisions on where to strategically place forces as a deterrent in a region. These methods also help to clarify critical variables and their associated uncertainties when viable.

When considering the framework of today's world, characterized by quick changes, volatility, and uncertainty resulting from the dynamics of the social, economic, and ideological variables that shape the operational environment, analysis and planning of potential futures (forecasting) allows decision makers to plan and act accordingly. These types of frameworks also provide a conduit for decision makers to see the possible results of decisions made with varying levels of certainty based on the modeled environments fidelity. One way of doing this is a relatively new approach to M&S called Agent Based Modeling (ABM).

2.7 Agent Based Modeling and Simulation (ABM/S)

ABM came about because of the desire to be able to accurately simulate real world scenarios, events, and interactions and the desire to gain a deeper understanding of complex, dynamic, and non linear behaviors amongst low level components. It uses a bottom up approach to create a model of complex systems that relies on creating simple "agents" and defining interactions in between these agents to generate a system that represents realistic behavior. ABM/S is primarily used to enhance understanding of what ordinarily is likely viewed as an environment with multiple entities, with different behaviors, and differing characteristics. Employing ABM/S often times affords a designer or decision maker a way explore options that under normal operating conditions could be detrimental, costly, or just plain infeasible. This technique, if setup properly, can accomplish several outcomes:

- Provide emergent behaviors that go unnoticed during real world activities.
- Allows a decision maker to see the impact or lack of impact of input variables.
- Allows the incorporation of personality traits of various entities to be modeled in one environment along with rules of interaction.
- Allows a decision maker to see potential evolution of an agent's ability to learn.

According to Lewe in [68] "the major strength of ABM/S comes from the fact that it is a simple, versatile, and flexible method that is well suited for studies of complex non linear systems." Biltgen provides a straight forward diagram of the concept of agent based modeling and it is reproduced in Figure 16 below.

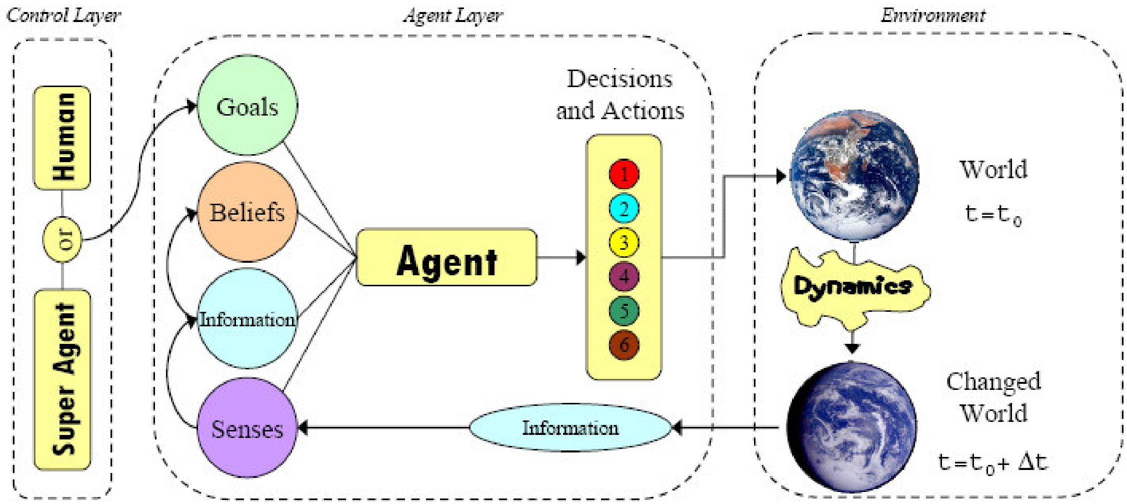


Figure 16: Flow Diagram of Agent Based Modeling [13]

ABM/S can be a very helpful tool in allowing decision makers to expend very little resources besides manpower to build a model and computing power to pose, examine, and propose solutions to a multitude of problems only limited by the creator's imagination. This capability can be very useful in providing support for a solution or family of solutions for problems as long as the model is representative of the problem environment and the assumptions inherent in the model are understood and valid.

2.8 Meta Models

As technology advances and computers become more capable, the models designers and engineers build will become more and more complex. While this does add fidelity to the modeling and simulation process it also adds complexity and computation time to the tasks of useful design space exploration and rapid assessments of the design environment. In the past, linear methods were used to approximate multi-dimensional curved spaces and that was the best one could do because computing power was limited. This is no longer the case and several methods have been developed over the years to create meta-models or models of a model to help facilitate design space exploration to reduce the design space so that detailed analysis or exploration can be accomplished without having to iteratively explore the entire design space. Two popular methods are Response Surface Equations and Artificial Neural Networks.

2.8.1 Response Surface Equations (RSE)

RSE's are a set of equations that attempt to model a complex system model that has several input factors that drive the responses or outcomes of a design. RSE's are referred to as quadratic regressions and are usually employed when simplification of complex and intricate models are sought. They capture the dependencies of responses or output metrics, to the independent input variables used to characterize or define the design space [64]. The process utilized to generate RSE's is called Response Surface

Methodology (RSM). “RSM is a set of techniques designed to find the “best” value of a designed response” [65]. The general form of a RSE is shown below:

$$R = b_o + \sum_{i=1}^{k} b_i k_i + \sum_{i=1}^{k} b_{ii} k_i^{\,2} + \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} b_{ij} k_i k_j + \varepsilon \qquad \text{Equation 4}$$

where R = the dependant parameter (response) of interest
bo = the intercept
bi = the regression coefficients for the first order terms
ki,j = the independent variables
bii = the coefficients for the pure quadratic terms
bij = the coefficients for the cross product terms
ε = error term

The process of formulating RSE’s was very difficult before the advent of the computer and software programs for statistical/DOE platforms. While useful, RSE’s do have their limitations. They can only be used in smooth, continuous, and well behaved design spaces; discontinuous spaces can not be modeled with RSE’s. RSE’s are also limited in the number of independent variables that can be considered before the model breaks down. Once the RSE has been generated and a good fit is determined, these meta-models can be employed for variety of analysis. One such capability is the ability to perform sensitivity analysis on input factors as compared to the responses. This is important because as the design space grows the ability of the designer to intuitively see variable effects on the responses diminishes [65]. RSE’s can also be used for optimization of the design space thus replacing complicated or costly running of codes for subsequent analysis.

2.8.2 Artificial Neural Networks (ANN)

It has been said the most advanced computer known to man is the human brain. It has been estimated that the cerebral cortex has 100 billion neurons [61]. Each neuron has

1000 dendrites (input path), hence 100,000 billion synapses (computer processing) [61]. Each neuron can fire at about 100 times per second, thus potentially executing 10,000 trillion synaptic activities per second [61]. It is almost logical then that to emulate the human brains capacity for problem solving along with complex computer programming would be beneficial in many ways.

Several distinguishing features of ANNs make them valuable and attractive for a forecasting effort. First, ANNs are data-driven self-adaptive methods. They learn from examples and capture subtle functional relationships among the data even if the underlying relationships are unknown or hard to describe [148]. Thus, ANNs are well suited for problems tasks whose solutions require knowledge that is difficult to specify, but for which there is enough data or observations. Second, ANN can generalize or postulate, even in the presence of noise, after being presented with a data sample [49]. This capability has shown the ANN capable of determining underlying relationships not always evident in typical analysis processes. Third, ANN's are universal function approximators; in that it has been shown that a network can approximate any continuous function to any degree of accuracy [49], [55]. Finally, ANN's are well suited for nonlinear problems and able to be applied to an increasingly large amount of real world problems previously unsolvable with conventional linear or linear approximation methods [49]. ANNs have shown the ability to provide acceptable surrogates for certain analysis activities and more information on the biological inspiration of ANN can be seen in Appendix D.

2.9 Monte Carlo Simulation

A Monte Carlos Simulation (MCS) is a very useful technique for performing uncertainty analysis. There is really no one correct method for conducting MCSs, but they all do tend to have the following steps to execute an experiment [89]:

1. Create a parametric model, $y = f(x_1, x_2, ..., x_q)$.
2. Generate a set of random inputs, $x_{i1}, x_{i2}, ..., x_{iq}$.
3. Evaluate the model and store the results as y_i.
4. Repeat steps 2 and 3 for $i = 1$ to n.
5. Analyze the results using histograms, summary statistics, confidence intervals, etc.

Monte Carlo methods generally are employed when it is not feasible or it is impossible to compute an exact result with a deterministic method [12]. Shonkwiler in [118] states that "Monte Carlo methods are used to simulate stochastic processes that occur in science, engineering, business and entertainment, to numerically solve problems in mathematics, science and engineering, and to ensure the security of data." MCSs model uncertain input variables with a probability density functions (PDFs) and involves running a large number of experiments in which the input values have been randomly sampled from the PDFs. This random sampling of each independent variable for each case that is simulated is usually generated from a uniform distribution, but other distribution shapes (Gaussian, exponential, etc.) can also be used if the designer is confident in the input variable's representative shape. The result of the experiments is cumulative distribution functions (CDFs), which are the result of taking the integral of the output PDFs. The CDF can then be used to quantify the ability to meet some requirement, need, or desire by the decision maker based on distinct input variable values.

Monte Carlo techniques are very difficult to employ where complex analysis is involved, such as time dependant or dynamic environments. For this reason, they are often combined with a meta-modeling techniques in which thousands of cases can be run on a simplified equation of the environment, like RSE's or ANN's; often times yielding a good probability distribution of the outcome that is very useful to decision makers.

2.10 Fuzzy Logic and Fuzzy Sets

So far as the laws of mathematics refer to reality, they are uncertain, and so far as they are certain, they do not refer to reality.

- Albert Einstein

The idea of fuzzy logic/fuzzy sets is not a new concept, it was originally introduced by Lofti Zadeh in 1965 as an attempt to deal with ill-defined problems (real world problems), often characterized by a certain degree of uncertainty and vagueness. The HITL decision maker is able to process generally at most nine pieces of information, often times that number is just seven [35]. "The past few years have witnessed a rapid growth in the number and variety of applications of fuzzy logic, ranging from consumer products and industrial process control to medical instrumentation, information systems, and decision analysis" [142]. Just about any form of engineering, mathematics, or science is now capable of being converted into a fuzzy system. One of the fundamental questions brought forth when trying to model the human problem solving process is, how do we learn? Another question of interest is how do we express ourselves? People communicate linguistically (with words), that is obvious, but what is the exact meaning of those words? Computing with words is a necessity when the available information is too imprecise to justify the use of numbers, and when there is a tolerance for imprecision which can be utilized to achieve tractability, robustness, low cost solutions, and or a better rapport with reality [15]. When a person looks at another and says he or she is tall or short there really is no clear definition, much of it comes from a person's perspective or inference with what the definition of tall or short is. One person may think tall is someone over 5'10", while another may think only anyone above 6'0" is tall. This discrepancy is apparent in many real world situations where imprecisely defined "classes" play an important role in human perception in the areas of pattern recognition, communication of information, and abstraction [144]. A fuzzy set is a class of objects with a continuum of grades of membership and is characterized by a membership function, $f_A(x)$, which assigns to each object a grade of membership ranging between zero and one [144]. The notions of inclusion, union, intersection, complement, etc from classical set theory are extended to fuzzy sets (extension principle). A classical set is a theoretical container that wholly contain or excludes objects based on their properties. For example if we had a bunch of words that were numbers and colors the two ensuing classical sets would be one of strictly numbers and one of strictly colors with no cross over. This theory was introduced by Aristotle who first formulated the Law of the Excluded Middle, which says X must either be in set *A* or in set *not-A* [78].

A fuzzy set, A, in X is characterized by its membership function, $f_A(x)$, on the interval of [0,1], with $f_A(x)$ representing the grade of membership. Reasoning with fuzzy logic is the "smudging" of the familiar yes-no (Boolean) logic where true (yes) equals one and false (no) equals zero; it also allows values such as 0.3 or 0.77. For instance if we were to take the dominant colors of the rainbow: red, orange, yellow, green, blue, indigo, and violet; these by themselves could be assigned values along a crisp (clearly defined) set of numbers to associate color definition. Looking at close proximity colors with respect to blue in Boolean logic the following questions would get the following response:

Q: Is green the color blue?
A: 0 (false or no)
Q: Is blue the color blue?
A: 1 (true or yes)
Q: Is indigo the color blue?
A: 0 (false or no)
Q: Is violet the color blue?
A: 0 (false or no)

But with fuzzy logic you would get the following answers:

Q: Is red the color blue?
A: 0 (false or no)
Q: Is blue the color blue?
A: 1 (true or yes)
Q: Is indigo the color blue?
A: .88 (for the most part yes, but slightly not blue)
Q: Is violet the color blue?
A: .56 (yes, but not a true blue)

So a plot one can expect of the crisp response and fuzzy response to the question of "blue-ness" theoretically would look like the one shown in Figure 17 below.

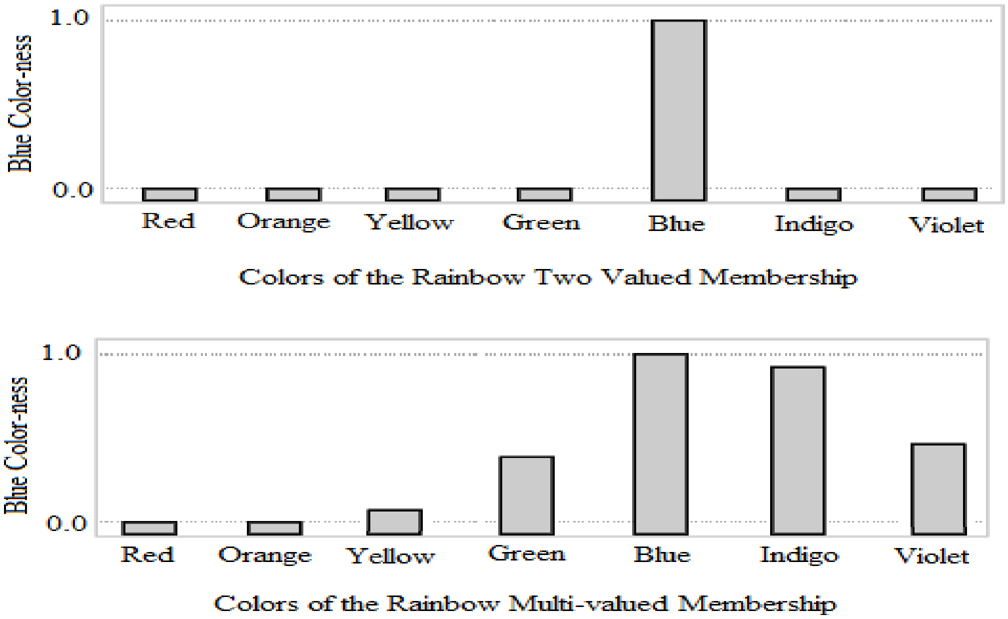


Figure 17: Example of Crisp versus Fuzzy Response

Now consider a continuous plot of Figure 17 shown in Figure 18 below.

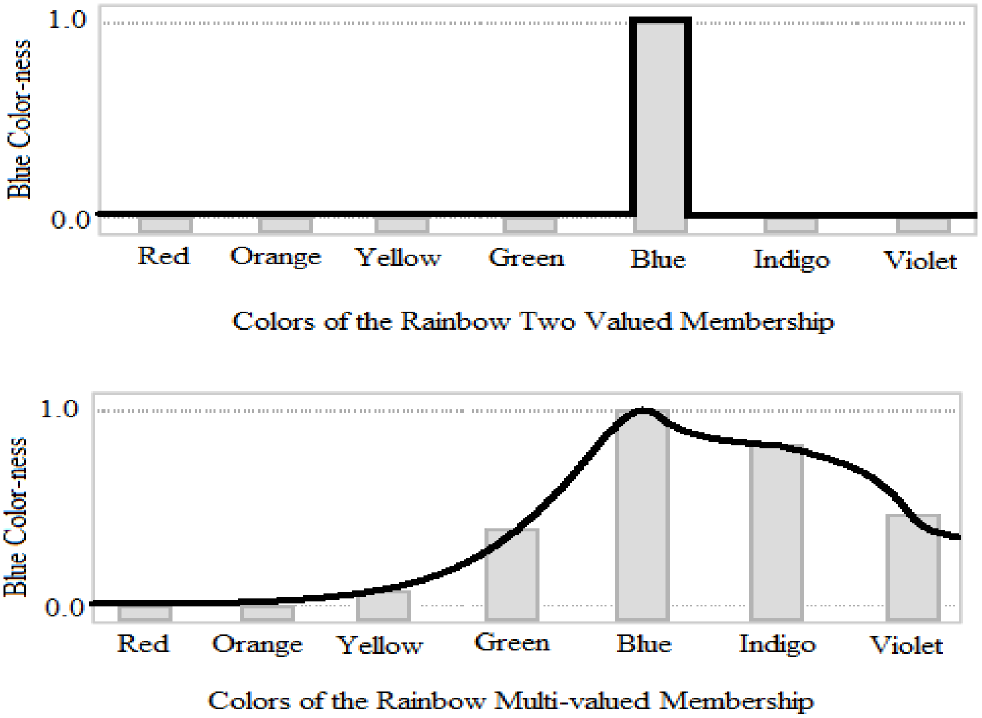
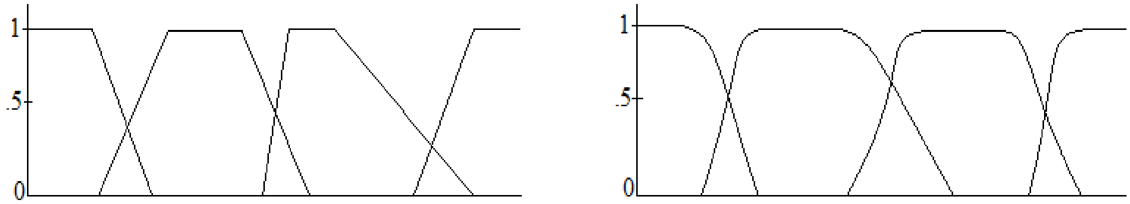


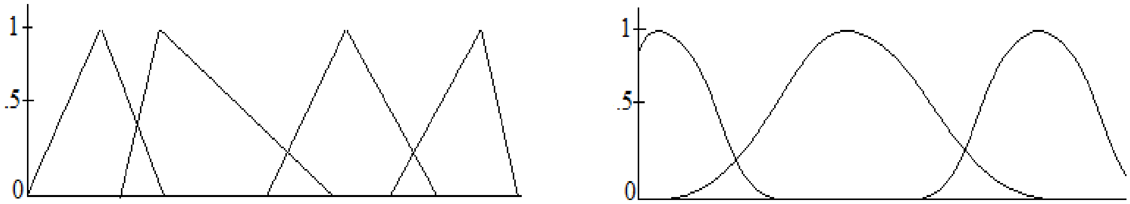
Figure 18: Example of Continuous Crisp versus Fuzzy Response

The plot on the bottom of Figure 18 shows a smoothly varying curve that accounts for the fact that all of blue and to a partial degree, yellow, green, indigo and violet partake some quality of blue-ness and thus deserve partial membership in the fuzzy set of the definition of blue. The ability to display this information in a continuous fashion instead of discreetly will have important implications especially in the area of taking the derivative. It should be noted that although the membership function may appear to look similar to a probability density function, there are differences and the fuzzy set is non-statistical in nature [144].

As was indicated in the example above, there is generally some overlap between fuzzy values, hence membership. There are several types of shapes available to the designer for characterizing input membership functions consisting of triangular, trapezoidal, Gaussian, exponential, polynomial, and so on. From the literature, a good rule of thumb for overlap is usually about 0.5, but can be less depending on the variable characteristics [105]. Determining this overlap is at the discretion of the designer. Examples of some of the shapes and corresponding overlap possibilities are below.



Trapezoidal Membership Function Polynomial Membership Function



Triangular Membership Function Gaussian Membership Function

Figure 19: Sample Fuzzy Logic Membership Functions

These types of overlap in the fuzzy value system can sometimes cause several rules to fire simultaneously with differing truth levels, each dictating a different output fuzzy set. To solve this dilemma, several resulting fuzzy sets can be integrated via an aggregation operation to obtain a reduced fuzzy set that describes the output of the end fuzzy system [105]. Generally the OR operator is used for this aggregation process.

Fuzzy Inference Systems (FIS) are the conduit for applying fuzzy logic and user defined membership functions. They generally consist of four main components. They are the knowledge base, which contains the fuzzy rules (rule base) and the database (membership functions), the fuzzifier (converts crisp values to fuzzy one), the inference engine that applies the fuzzy reasoning input by the designer, and a defuzzifier, which can take fuzzy output and translates into a crisp output. A diagram of a FIS can be seen

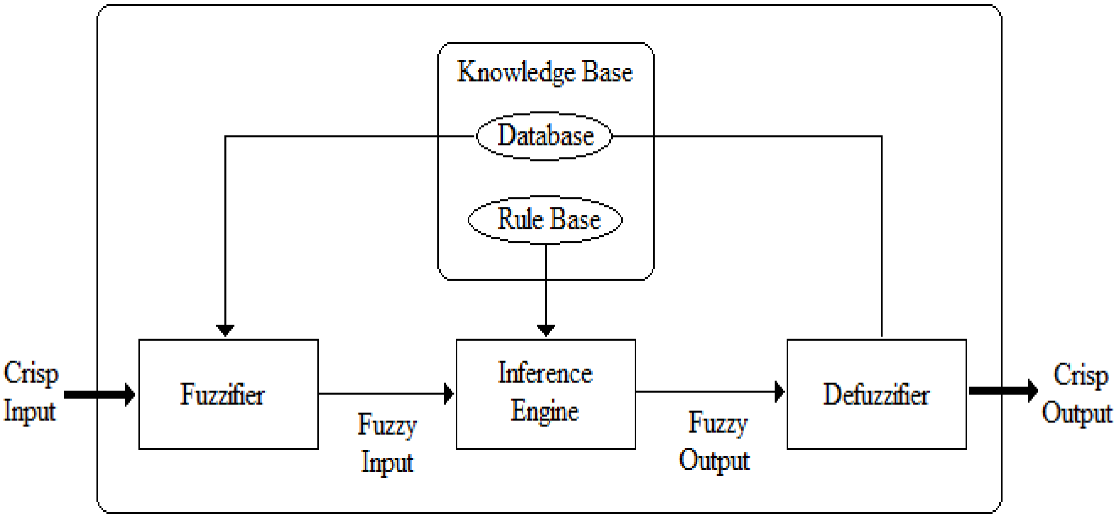


Figure 20: Structure of a Fuzzy Inference System [105]

in Figure 20. The key steps to solving a fuzzy problem are as follows [61]:

1. Understand and define the problem in detail.
2. Determine the variables and their ranges.
3. Apply appropriate membership function profiles to each variable.
4. Determine IF-THEN rule statements and using logical operators if necessary.
5. Select an appropriate de-fuzzification methodology.
6. Test the system to verify correct outputs; if needed go back to step 3 till error is within acceptable limits.

Focusing back now on the linguistic aspect of fuzzy sets, Zadeh in [145], [146] and [147] goes on to describe the use of linguistics as a variable. The reason for the direction in utilizing more qualitative statements than quantitative data is the ineffectiveness of computers to deal with or emulate humanistic systems. Zadeh calls this phenomenon the *principle of incompatibility* [145]. This principle asserts that high precision is incompatible with high complexity. Meaning that conventional techniques and mathematical abstractions are incapable of accurately modeling the complexity of the human thought process and decision making criteria. Thus an exploration into an area of less rigor in definition of an environment is needed and born was the idea of *approximate reasoning*. The definition of *approximate reasoning* is “a type of reasoning which is neither very precise nor very imprecise” [145]; very middle of the road, but as the reader should begin to understand very necessary for applications towards real world problems.

A linguistic variable is structured in such a way that it is associated with two rules; they are the *syntactic rule* and the *semantic rule* [146]. The *syntactic rule* specifies the way in which the linguistic values which are in the term-set of the variable may be created. The term-set is the totality of values a linguistic variable is capable of having. With regard to this rule, the assumption is that the terms in the term-set of the variable are generated by a context-free grammar [145]. The *semantic rule* specifies the procedure for computing the meaning of any given linguistic value. In this context, the use of a typical value of a linguistic variable, e.g., *not very X and not very Y,* involves what are called the *primary* terms, i.e., *X* and *Y,* whose meaning is both subjective and context-dependent [146]. We assume that the meaning of such terms is *a priori.* Besides these *primary* terms Zadeh also refers to connectives such as *and, or, either, neither,* and the negation *not;* as well as *hedges* in the form of *very, more or less, completely, quite, fairly, extremely, somewhat,* etc. [145]. These three types of modifications to the primary term may act as operators that modify the context of their operands in a context independent fashion [145]. Finally, Zadeh refers to the fact that the base variable acting in a fuzzy fashion is linked to a suitable interval that can be linked via a compatibility function; i.e. if the linguistic variable is *very tall*, there will be some height that is associated with this linguistic and associated with some membership function [147]. In other words the compatibility function is based on a set of non-explicit impressions generally only cognitive to a human and not directly to a computer, which generally requires a set of well defined mathematical or numerical objects. For a more formal and mathematically intensive explanation of a linguistic variable and its applications, the reader is directed to [145] , [146], and [147].

The main advantage of the adoption of fuzzy logic is the ability to take vague concepts or linguistic expressions and use them to perform calculations. The use of fuzzy variables can be very instrumental in solving problems where HITL decision-makers are required. To this extent, fuzzy logic makes it possible to reproduce the sometimes ambiguous and subjective way of thinking conducted by human beings. There are many aspects of the fuzzy systems from fuzzy rules, fuzzy probabilities, and others not discussed in this section like fuzzy graphs. In an essential way, these methodologies reflect the fact that imprecision and uncertainty are pervasive and precision and certainty carry a cost. “In the final analysis, the principle aim of these methodologies is to exploit the tolerance for imprecision and uncertainty to achieve tractability, robustness, and low solution costs” [142].

2.11 Fuzzy - Neural Networks

With the advancement of technology in the late 1980’s and early 1990’s there has evolved ideas and concepts that seem to be born right out of the movies. The advancement of widely available electronic consumer products such as washer machines, microwaves, ovens, digital cameras, cell phones, and digital entertainment systems has incorporated extremely innovative capabilities. These features allow the consumer to basically just plug and play these products, meaning that users need to know very little about their operation for them to work. Today the machines contain extensive logic that allows them to operate almost autonomously. There are microwaves and ovens that will adjust the cooking temperature and cook time based on inferences of what stage of the cooking process an item might be in or by measuring the moisture in the air. There now exist music systems that track a users preferences in music genre and "learns' to play not just the particular songs selected, but others in a similar class. There are digital cameras that focus in on particular objects and even eliminate the slight movements of the human hand to create clearer and crisper picture quality. The list goes on and on. What has made this possible? Zadeh refers to this enabler as the Machine Intelligence Quotient (MIQ). He states in [143] that MIQ has resulted from the move away from the rigors of mathematically intensive computing to an area of *soft computing*. Soft computing is the idea that that precision and certainty carry a cost and that computation, reasoning, and decision making should exploit--wherever possible the tolerance for imprecision and uncertainty [143].

The principle ingredients of soft computing primarily consist of the theories of fuzzy logic, neural networks, and probabilistic reasoning with the last proliferating the areas of genetic algorithms, learning theory, and chaotic systems. Fuzzy logic is primarily concerned with the imprecision in real world problems, neural networks seek to provide a learning environment in which to classify and identify patterns or underlying causal relationships, and probabilistic reasoning seeks to deal with the inherent uncertainty associated with real world designs and problems. The inherent overlap in these three areas is readily apparent, but what is intriguing to many engineers and designers who use these methodologies that these once individually used methods are not competitive with one another, in fact they are complimentary [143]. By this definition it is meant that the inherent drawbacks of one method can be overcome or minimized by incorporating the pros of the others. Examples of this are the increasing use of neuro-fuzzy networks that are being created for the purposes of learning and adaptation to varying applicable environments.

Probabilistic reasoning is a new area not yet discussed, so a basic overview of this topic is warranted. The general approach to finding solutions to problems is normally dealt with assertions and their causal relationships. For example if we know someone has a fever and has the flu, then the assertion is: if somebody has the flu then that person has fever. Although this assertion seems true, it cannot be expressed with 100% certainty. There is some degree of belief or disbelief associated with the assertion. The problem is how to associate a degree of belief or of disbelief with assertions or [104]:

- How do we associate beliefs with elementary assertions?
- How do we combine beliefs in composite assertions from the beliefs of the component assertions?
- What is the relation between the beliefs of causally connected assertions?

Estimates for elementary assertions can be obtained from experts (subjective probability) and from frequencies (if given enough data) [122]. It is very hard to come up with good estimates for beliefs, one should always consider the question: *"What if the guess is bad?"*. Estimates are needed, given the belief in assertions X and Y, for the assertions complement, union and intersection of X and Y. Evidence must be combined in cases such as [122]:

- We have a causal connection from assertion X to assertion Y, what can we say about Y if X is true, or, vice versa, about X if Y is true.

- We have a causal connection from assertion X to assertions Y_1 and Y_2, what can we say about X if both Y_1 and Y_2 are true.

- We have a causal connection from assertion X_1 to Y and a causal connection from X_2 to Y, what can we say about Y when both X_1 and X_2 are true.

Probabilistic reasoning usually starts with a multivariate model that contains a set of variables Ω and a defined probability distribution $P(\Omega)$ [122]. The goal is to measure some hypothesis A with the aid of some evidence B [46]. This can be represented by the following equation:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} \qquad \text{Equation 5}$$

An important benefit of probabilistic reasoning is the fact that it is non-monotone with respect to the available knowledge. As a consequence, even if a hypothesis is almost perfectly likely to be true {false}, it may later turn out to be false {true} [46]. If the reader is interested in learning more about probabilistic reasoning they are referred to [94] and [117].

When one designs a neural network alone, the structure dictates that it is essentially a “black box” environment, meaning there is not much transparency in the logic used to come up with an answer. Inputs go into the environment and logical outputs come out. Fuzzy systems require an understanding of the fuzzy input variables and their associated membership functions. The input to output relationships as well as the good judgment by the designer to select the appropriate fuzzy rules that will contribute most to the solution of the problem or design being investigated is also critical. The curse of dimensionality is something many engineers are familiar with when dealing with problems with multiple variables with varied settings [48]. For example consider a system that has five inputs, with three settings each and one resultant output; the total number of rules in a fuzzy system would be 3^5 or 243, and for 6 inputs it becomes 729. This is a large number of rules, and the likelihood that many significantly impact the solution to the problem is low. Much research is currently under way that is focused at minimizing this issue.

Fuzzy neural networks are gaining popularity in industry, research, and academia for their abilities to help a designer easily model real world problems and apply a useful level of imprecision inherent in these problems. While fuzzy systems and neural networks both have the ability to work with imprecision and where the space is not defined by crisp, deterministic boundaries, they do have their downfalls. Fortunately, the drawbacks of neural networks and of fuzzy systems can usually be overcome if fuzzy logic operations are incorporated in neural networks and learning and classification accomplished by neural networks is implemented into fuzzy logic [61]. For more information on this subject and its underlying principles the reader is directed to Appendix D.

2.12 Decision Support System (DSS)

Decision Support Systems were originally developed to help aid decision makers in the decision making process in the early 1970's [70]. These tools, systems, or methods allow a DM to scope and scale the problem they are faced with into a more manageable data space. DSS usually incorporate a set of procedures which aid in expanding the cognitive space of the DM through use of a computer that is capable of accessing relevant databases or knowledgebase's. The best definition of a DSS is given by Adelman [2]:

> *Decision Support Systems are interactive computer-based systems (software), which use analytical methods such as decision analysis, optimization algorithms, etc, in order to develop appropriate models that will support decision makers in the formulation of alternative solutions, the resolution of the reactions amongst them, their representation, and finally in the choice of the most appropriate solution to be implemented.*

Thus, DSS are systems that attempt to leverage analytics, expertise, models, etc. that can be used to define or visualize an environment that is likely complex, nonlinear, and or ill-defined. Quality in decision making is generally the result of these systems and also can provide justification for specific COA selection or reasoning for a decision. These tools help alleviate the burden of data processing and management for a DM and have the potential to provide visibility to multiple users involved.

2.13 Literature Search of Tools

The preceding methods and techniques are often times applied or incorporated into tools that have been created to meet a specific need or have been designed to be used for a certain class of problem. This section will give an overview of several methods and applications they have been utilized in to provide the reader applications to forecasting analysis, supply versus demand, and how surrogate applications can be incorporated.

2.13.1 Auto-Regressive Integrated Moving Average (ARIMA) models

This method is a time series forecasting method that is sometimes referred to as the Box-Jenkins method. The model consists of three parts consisting of an autoregressive (AR) portion, the degree of differencing involved, and a moving average (MA) calculation [74]. In this fashion the model can also be referred to as the ARIMA(p,d,q) model where p is the order of the autoregressive part, d is the degree of differencing, and q is the order of the moving average process [6]. It is a generalization of the autoregressive moving average (ARMA) model. Both are given a time series set of data, X_t, and attempt to better understand the data or to predict future points in the series. There are several models in ARIMA to choose from such as the ARIMA(0,0,0) and ARIMA(0,1,0) [74]. The ARIMA(0,0,0) is a simple random model made up of two parts, an overall mean, μ, and a random error component, e_t, which is independent from period to period. Its equation is as follows [74]:

$$Y_t = \mu + e_t \qquad \text{Equation 6}$$

The non-stationary random model, ARMIA(0,1,0) is where the observation, Y_t, depends on the previous observation Y_{t-1} and is represented by the following equation [74]:

$$Y_t = Y_{t-1} + e_t \qquad \text{Equation 7}$$

The figure below shows illustrations of the time series data for these two processes.

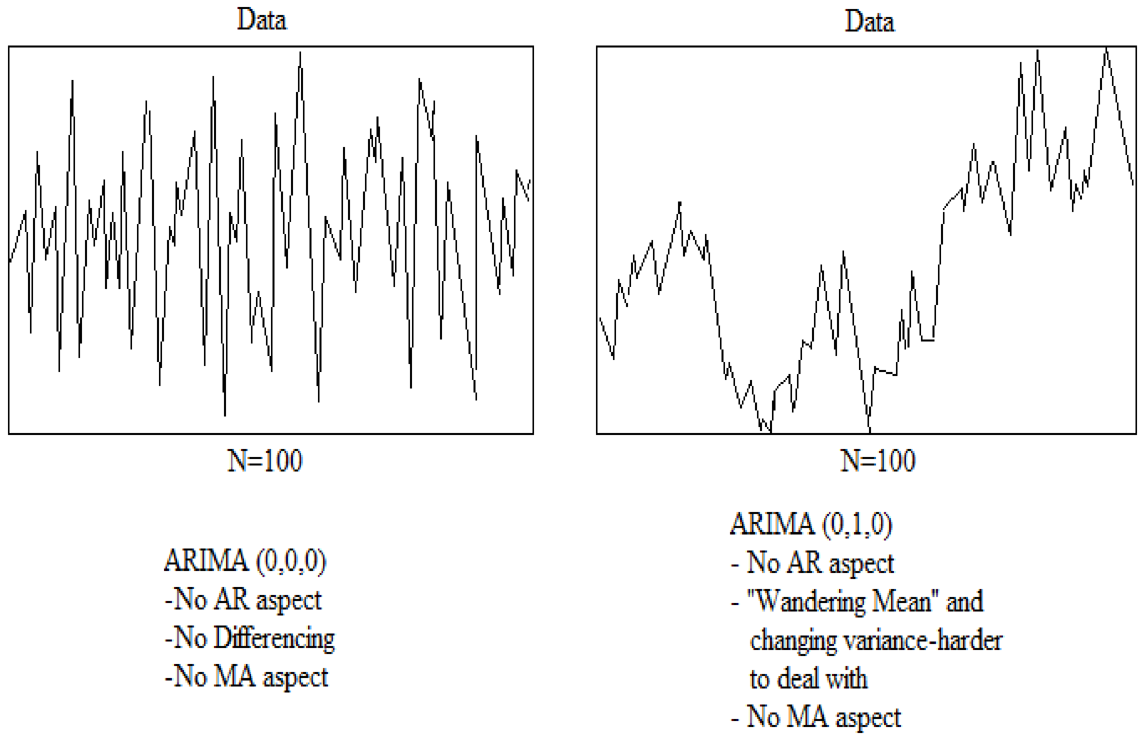


Figure 21: Time Series Data Examples for ARIMA models [74]

2.13.2 News Vendor Model

This model is utilized in the manufacturing field to help predict components or resources in which there is only one production or procurement opportunity [17]. Because that opportunity occurs well in advance of its intended use or selling of a product, the entire order is received just before a selling season or the usage period in question starts. There is a fixed cost per unit ordered and generally the decision made, based on predicted demand, will either leave you with all your goods sold or delivered; i.e. demand exceeds quantity ordered. Or demand will be less than the quantity ordered and leftover goods will be the result. Either way a loss is incurred through lost profits or increased costs due to extra inventory that will likely have to be written off or sold at a reduced rate. The goal is to forecast and predict exactly what is needed by matching supply with demand. The basic premise of this model is to balance the cost of ordering too much against the cost of ordering too little [17]. This is accomplished by using a combination of intuition and data analysis to construct the forecast. The intuition part comes from expert opinion and familiarity with the demand environment. The data analysis can come from a variety of source, but generally historical usage in a similar environment.

For example, U.S. forces are involved in a joint strike force operation with a small enemy force in a relatively warm climate. A logistician familiar with the usual need for food, water, ammunition, etc. for the combat force involved might make an initial forecast based on a set standard load or need requirement deemed adequate for a soldier per day for the number of days the operation is expected to take. His intuition might make him order more water and food than the standard per day basic load because of the physical demands placed on those soldiers in such a climate. The data analysis would come from past forecasted usage and time on the ground, where the actual usage of resources and time on the objective for a similar operation would be drawn upon for the current determination of logistics. This error can be measured with the relative forecast error of *A/F ratio*:

$$A/F \quad ratio = \frac{Actual \;\; Demand}{Forecast}. \qquad \text{Equation 8}$$

In this sense, it is more important to look at the relative error instead of the absolute error, represented by:

$$Error = \frac{Actual - Forecast}{Actual} \text{ [18].} \qquad \text{Equation 9}$$

When several key or critical items are being analyzed, visibility of these items is not lost in the noise with the *A/F ratio*. For the news vendor model, an accurate forecast is indicated by an *A/F ratio* of 1, above one is too low of a forecast, and below 1 is a forecast that is too high [17].

If given the previous operations *A/F ratio*, it is not infeasible for the logistician described earlier to think the two should be comparable and to base his new forecast on those errors. Utilizing standard results from statistics and Equation 8 we get the following results:

$$Expected \;\; Actual \;\; Demand = \mu = Expected \;\; A/F \; Ratio \times Forecast, \qquad \text{Equation 10}$$

and

$$Std\,Dev\,Actual\,Demand = \sigma = Std\,Dev\,A/F\,ratios \times Forecast \qquad \text{Equation 11}$$

Utilizing a normal distribution, the mean and standard deviation are now apparent. Next the logistician could go to a standard normal distribution function table or use a software package to obtain the distribution function, Φ(z). It is highly unlikely that the demand forecast will be a standard normal distribution, but it can be used to find the probability that the demand, *F(Q),* where *Q* is the value in question, by converting *Q* into an equivalent quantity for the standard normal [17]. In other words find a *z* such that *F(Q)* = Φ(z). This is called the *z statistic* and is calculated with the following equation [17]:

$$z = \frac{Q - \mu}{\sigma} \qquad \text{Equation 12}$$

Again the idea is to find the probability demand is less than or equal to Q. Suppose the value of *z* is equal to 0.79, therefore the distribution function is Φ(0.79), which from the "z table" gives a probability of 0.7852 or just over a 78% chance the demand will be *Q* or less.

As stated before one of the benefits of this model is the balancing of the cost of over or under ordering. In the scenario presented, if an overage cost, C_o, and an underage cost, C_u, could be quantified we could set the expected loss of the Q^{th} unit to the expected gain on the Q^{th} unit. The resulting equation would be [17]:

$$C_o \times F(Q) = C_u \times (1 - F(Q)). \qquad \text{Equation 13}$$

Rearranging terms would result in the following:

$$F(Q) = \frac{C_u}{C_u + C_o} \qquad \text{Equation 14}$$

Thus we could calculate an optimal order quantity, *Q*, or needed resources for the operation by matching it to the calculated percent probability that demand needed will be *Q* or less. The decisions made by the military logistician in the above scenario is one where soldier's lives are at stake, not profits or monetary loss, so a deeper understanding of the impacts of an ordering methodology similar to the news vendor model would need to be examined closely before any implementation.

2.13.3 Fuzzy Cooperative Co-evolutionary (CoCo)

This methodology applies the concept of two coevolving species that are defined as the database or membership function and the rule base. This model is primarily based on the framework defined in [106] and [107]. It is designed to allow the user to manage the trade-offs between performance and interpretability [105]. The fuzzy logic process entails receiving fuzzy inputs which can consist of linguistic variables, which then must be associated with the proper membership function while at the same time identifying the connective parameter or rules used to execute the fuzzy inference system. Consequently the first species known as the operational parameters will be used to encode the values which completely define the membership functions for all the variables of the fuzzy system [105]. The second species defines the rules set in the form *if* (x_1 is Y_1) *and …and* (x_n is Y_n) *then* the *output* is Z [105]. Finally a genetic algorithm can be used to control the evolution of the two species as a population [139].

A genetic algorithm applies three phases of selection in an attempt to find the best and strongest outcome. The first phase assigns a fitness level in order to select the mating pool. It has an elitist strategy that only allows the best individuals to move on to the next generation. The second phase is called the standard crossover and mutation phase of the algorithm. This is where crossover between two genomes is executed with an associated probability by selecting at random a single crossover point and exchanging the subsequent traits to form two new offspring [87]. If no crossover takes place the genomes are passed on as they are. The third phase, mutation is performed by flipping bits in the genome structure according to some predefined probability of mutation. The algorithm terminates when some level of fitness is realized or the maximum number of iterations is reached.

2.14 Modeling & Simulation Tools for Military Applications

In 1997 the United States Congress authorized an applied research project to evaluate non traditional combat simulation techniques that would potentially be useful in addressing three areas that were largely omitted in existing combat simulations. These three areas were and are defined as [14]:

- *Nonlinearity*: Small changes in inputs that have disproportionate effects on outcomes.
- *Intangibles*: Human factors, effects of leadership, troop morale, and other difficult to define aspects of combat.
- *Coevolving Landscape:* The premise that adversaries continually anticipate one another's actions and base their decision on the prediction.

With those three research objectives in mind an effort quickly developed into an international consortium involving eight other countries to see how best to approach modeling combat more effectively and accurately. This section discusses several tools developed to facilitate a solution to the issues above.

2.14.1 Logistic Planning in the Armed Forces Today - OPLOGPLN

In an effort to reduce some of the uncertainty inherent in logistics forecasting and planning, the military has created a wide variety of tools for the logistician. One such tool for the Army, generally the largest ground force in a conflict, is a program called Operations Logistics Planner (OPLOGPLN), currently at version 7.0 (released December

2007). OPLOGPLN is meant to assist the logistics planner in calculating logistics in support of any user defined operation. This tool is specifically designed to support operations typically associated with multi-phase Operation Plans (OPLAN) and Operation Orders (OPORD). The user creates units based on standard Tables of Organization and Equipment (TOE) and maps these units into Task Organizations (TO). The TO can then be assigned to a multi-phase order and assigned user-developed mission parameter sets (which essentially describe the conditions under which the TO will be in or is operating in). To narrow down the scope of information only the requirements for the nine classes of military supply are calculated. These classes are described in Table 3.

Table 3: Military Class of Supply System

Supply Class	Description
Class I	Subsistence (food) and water in packaged form or served "fresh."
Class II	General supplies that can consist of troop support material items such as clothing & textiles, weapons, and a variety of industrial supplies.
Class III	Petroleum, oil, and lubricants (POL) in packaged and bulk form to fuel equipment and maintain it.
Class IV	Barrier and construction material for a variety of requirements like base construction material, defensive positions, obstacles, etc.
Class V	Ammunition in the form of bullets for a variety of weapons, grenades, explosives, etc.
Class VI*	Sundries supplements, health and comfort packs, and personal demand items.
Class VII	Major end items i.e. vehicles, generators, aircraft, etc.
Class VIII	Medical material from the soldier level to hospital level.
Class IX	Repair and replacement parts like batteries, vehicle parts, weapon parts for repairs, etc.

This tool will be discussed further in Section 5.2.1 in concert with a proof of concept scenario that compares OPLOGPLN and the ATLAS method developed by the author to address the gap identified by the military for adequate logistics requirements and architecture determination.

2.14.2 Analogical Hypothesis Elaborator for Activity Detection (AHEAD)

As previously stated several organizations in the defense and commercial sector are attempting to solve the problem of forecasting or predicting catastrophic events, especially in the post 9/11 era. Conducting intelligence analysis is one of the primary activities that will allow the US to respond accordingly to perceived threats of our national security and those of our allies. Terrorism awareness has taken center stage in this era and will likely remain that way for the foreseeable future. Terrorist and their activities can be characterized as asymmetric threats because the tactics employed generally consist of varying forms of guerilla warfare. This strategy has always served the “little guy” in a conflict well because it requires less infrastructure, allows distributed command and control, and although a goal may be unified (i.e. a jihad against the west) it is generally so broad that any activity that supports that goal is sought by its participants. Unfortunately early detection of these aggressors has proven to be a difficult task for many threat domains because usable data sets tend to be large and complex. This is generally due to the relationships involved among the variables that define the threat landscape [91]. For this reason and the acceptance that the asymmetric threat domain has relatively infrequent instances and is sufficiently complex to encompass an infinite amount of variations, the Navy Center for Applied Research in Artificial Intelligence along with the Defense Advanced Research Projects Agency (DARPA) has developed a tool called the Analogical Hypothesis Elaborator for Activity Detection (AHEAD). This tool's primary purpose is to increase the level of detection of potential terrorist threats through use of information and intelligence analysis via user interaction.

AHEAD employs the principles of case based reasoning (CBR) to give a threat hypothesis which is directly tied to a relevant piece of evidence and then focuses on elaborating on that hypothesis with justification on why that hypothesis is supported or not by an analyst's assessment. One of the differences from traditional CBR is that AHEAD's cases by design do not need to be structured for efficient matching of large bodies of evidence [91]. However, these cases do need to include what kinds of evidence are consistent with the proposed hypothesis as well as justification that the evidence supports the hypothesis. This in effect generates arguments as to why a specific threat could be occurring in the detection process. The structure of AHEAD specifically concentrates on relating facts from the evidence to specific assertion about the process being performed [90]. AHEAD's algorithm consists of three elements, they are described below [90]:

1. Retrieve: Given a proposed hypothesis and library of cases representing the activity being investigated, retrieve the model that best matches.
2. Propose: Given the above matched model and evidence, generate the instantiation that best matches the evidence. Instantiation occurs because the cases used by AHEAD are generalizations of actual cases.
3. Justify: Given the logic and evidence, analyze the situation described by the evidence and create arguments (for and against) explaining why a situation does or does non match a case.

Figure 22 provides the flow process and basic functional architecture developed to provide information to a data analyst seeking to collate data and arrive at a suitable threat assessment based on the inputs.

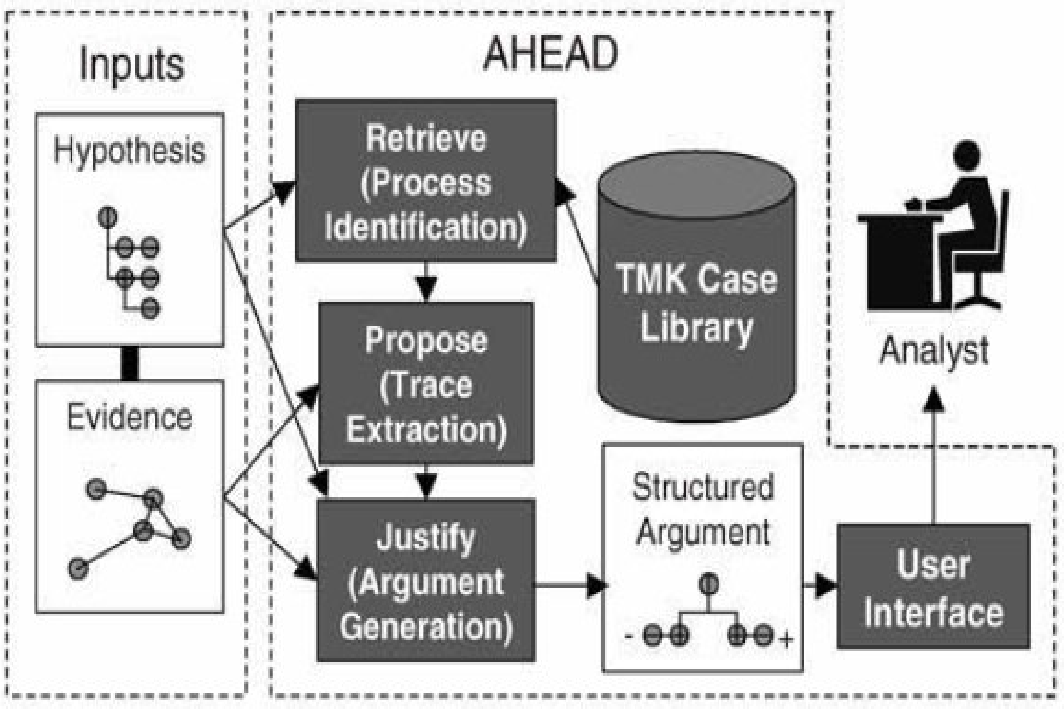


Figure 22: Functional Architecture of AHEAD [90]

To facilitate the execution of AHEAD a graphical user interface (GUI) was developed to allow a user to browse through the semiformal and informal arguments associated with each hypothesis. This is accomplished by allowing exploration of the tree consisting of different hypotheses and arguments, which in turn are linked to the argument's evidential support. Tests performed using this methodology and tool showed on average a reduction in decision making time by 10%, an increase in confidence that the conclusions reached by an analyst with AHEAD were sound and justifiable [90]. Results also showed a reduction in error in judgment and confidence by about half when considering a decision made with or without the arguments from the case database was also observed [90]. These are encouraging results as asymmetric and complex environments that were once almost impossible to predict are now being dissected with methods from the cognitive sciences and CBR techniques to provide insight to users.

2.14.3 Enhanced Irreducible Semi-Autonomous Adaptive Combat (EINSTein)

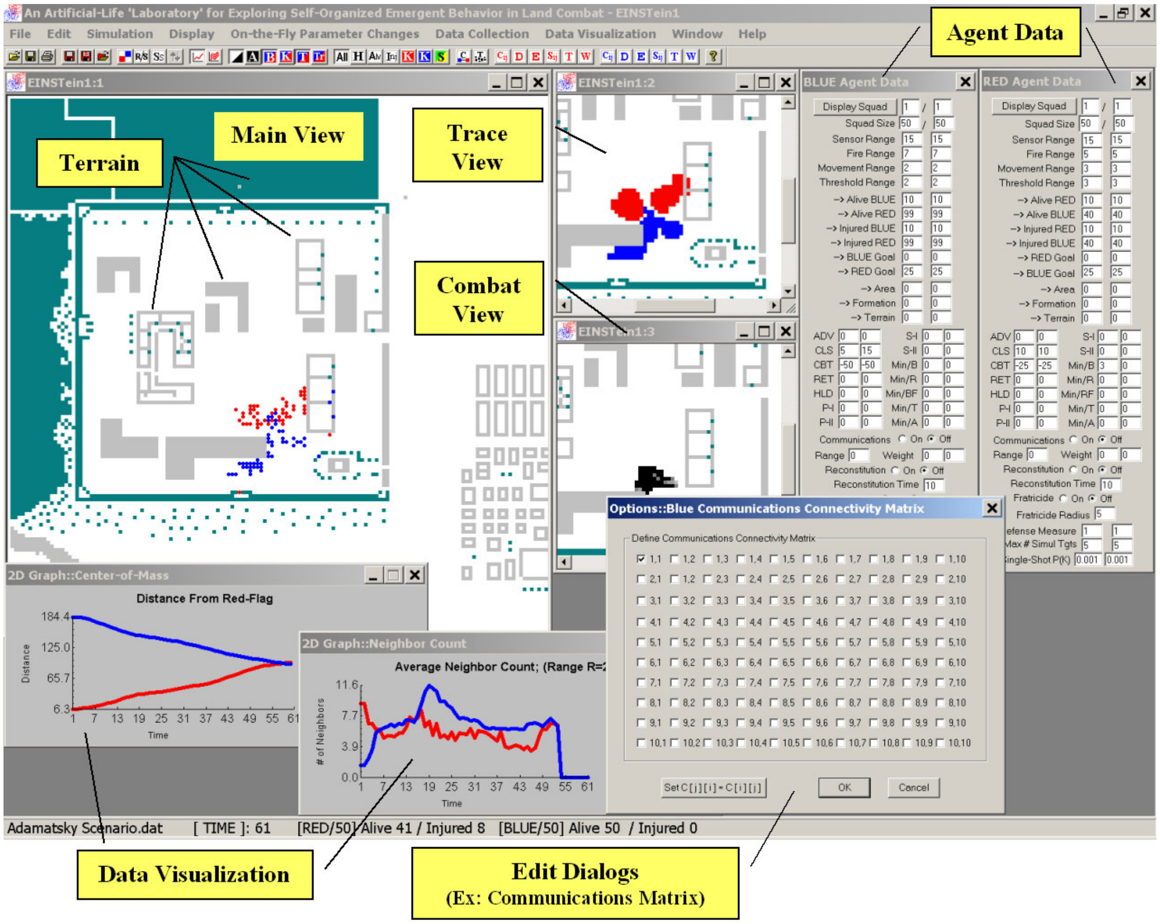
Agent based simulation as discussed earlier is increasing in usage for a variety of industries and their decision makers so that conducting *what-if* analysis can be used to visualize potential outcomes. The USMC has a vision of future combat: "small, highly trained, well armed autonomous teams working in concert, continually adapting to changing conditions and environments" [53]. This vision initiated a transformation to a different force strategy and architecture discussed in Chapter 1 called Distributed Operations. In the later 90's the program called Irreducible Semi-Autonomous Adaptive Combat (ISAAC) was developed by the Center for Naval Research and Analysis. The core dynamics of ISAAC is patterned after mobile cellular automata rules [54]. These rules take a bottom-up approach to the modeling of combat, versus the more traditional top-down, or reductionist approach. ISAAC utilizes agents or ISAACA, which represents a primitive combat unit (infantryman, tank, transport vehicle, etc.) that is equipped with the following characteristics [53]:

- Doctrine: a default local-rule set specifying how to act in a generic environment
- Mission: goals directing behavior
- Situational Awareness: sensors generating an internal map of environment
- Adaptability: an internal mechanism to alter behavior and/or rules.

A global rule set determines combat attrition, reconstitution, and reinforcement. ISAAC also contains both local and global commanders with their own command influence radius and obeying an evolving C^2 hierarchy of rules [53]. Initial ISAAC proof of concepts performed well and spurred on further development of trying to model combat through ABM, ergo EINSTein.

EINSTein is an adaptive ABM and simulation tool for modeling combat operations on a small to medium scale by using autonomous agents to model individual behaviors and personalities instead of specific weapon effects [52]. The purpose of creating EINSTein was to investigate and see if there existed conditions under which high level patterns (penetration, flanking maneuvers, feint, etc) of emergent behavior could be gleamed from a set of low level actions (move forward, backwards, retreat, attack, etc.) [52]. Hence, EINSTein is meant to be used as an interactive tool in which a user can explore the high level emergent behaviors and explore the middle ground between highly complex and realistic models. This ABM seeks to overcome the limitations of the simplistic model that contains only the minimum dynamic variables that fail to create interesting real behaviors of combat entities and provide little insight to basic processes [52], [53].

EINSTein consists of three modules. These are the combat engine, the GUI, and the data collection / data visualization functions. The combat engine processes all the decisions concerned with combat and is the core script for which multiple time series data collection, fitness landscape of the agents in their operational space, and associated algorithms all depend [52]. The agents in EINSTein have the same characteristic described earlier for ISAAC and exist in one of three states: alive, injured, or killed. Characteristics can be applied to agents for different states. For example, an agent in the *alive* state performs normally, while an agent who is *injured* could have its effectiveness reduced by some user defined percentage to account for the injury. The battlefield is represented by a two dimensional lattice of discrete points in which only one agent may occupy a point at a time. Personality traits are defined by a vector of six weights that characterize the agent's reactions in the presence of other enemy or friendly agents within its perception range. A screen shot of EINSTein's GUI and data entry dialogues is shown in Figure 23.



Adamatsky Scenario.dat [TIME]: 61 [RED/50] Alive 41 / Injured 8 [BLUE/50] Alive 50 / Injured 0

Figure 23: Screen Shot of EINSTein Simulation Tool [53]

2.14.4 Map Aware Non Uniform Automata (MANA)

MANA was created by the New Zealand Defence Technology Agency for the purpose of conducting military agent based modeling and simulation. Its development is inspired by chaos and complexity theory and was developed in conjunction with the Marine Corps' Project Albert initiative [109]. Similar in thought to EINSTein, the designers of MANA wanted to incorporate a way to define, at least in some part, many of the intangibles experienced across the battlefield. These intangibles consist of but are not limited to situational awareness (SA), information edge provided by enhanced sensor capability, and certain aspects of command and control [82], [83]. MANA is based and has been developed based on two foundational ideas [83]:

- The behavior of the agents within a model is a critical component of the analysis of possible outcomes.
- That decision makers are wasting their time with highly detailed physics based models for determining force mixes and combat effectiveness.

Because human nature is relatively a mathematically intangible variable it is often overlooked or unjustly simplified by modelers. This decision often creates models that even to the lay person outputs results that make very little or no sense, only to be "adjusted" to represent reality. This tool has been used extensively not only in M&S for the New Zealand Army's research, but it has been used in several Naval Post Graduate School theses to analyze real military world problems to include humanitarian aid operations, unmanned aerial vehicle deployment, and convoy protection operations [8].

MANA provides a venue for the user to model complex combat situations that include terrain features, a variety of individual agents with their own set of personality traits, and varying communication and or command and control capabilities. The primary MANA Parameters allow a user to specify [83]:

- Personality weightings that determine an agent's propensity to move toward or away from an influence on the battlefield (friends, enemies, waypoints and/or desirable terrain).
- Movement constraints that modify how an agent moves and are not covered under the scenario's personality weightings.

- The ability to define a variety of agents, weapons, sensors, movement speeds, targeting priorities, and communication assets.
- Algorithm modifiers that define whether terrain affects speed, degree of randomness when moving, whether obstacles should be avoided, and the specified movement algorithm for the agents used in the model.

Several runs of a model can be executed in relatively a short amount of time, depending on its complexity. MANA also enables exploration of outcomes via different initial inputs or character settings through execution of a design of experiments (DOE). This is facilitated by the use of a structured computer code language called extensible markup language (XML) that provides a formatted file that can be passed to MANA. MANA also has the capability of interfacing with CONDOR, a program developed at the University of Wisconsin that allows computer resources that are common to a network to share their distributed computing power to increase execution and run time of a single networked computer's processing activities. An example of this was displayed by Bain in [8] in which his experiment would have taken approximately 700 CPU days to execute or almost two years to complete on one machine, but he was able to complete his experiments in a month's time with CONDOR and 12 dedicated computers.

2.14.5 Pythagoras

Pythagoras is a combat modeling and simulation tool developed by Northrop Grumman. Mathematically measurable phenomenon like lethality, rates of fire, rates of movement are all physical aspects that can be represented in this modeling environment. However as shown in Figure 24 the combat environment is not only made up of physical world, it also encompasses the effects of leadership and human factors.

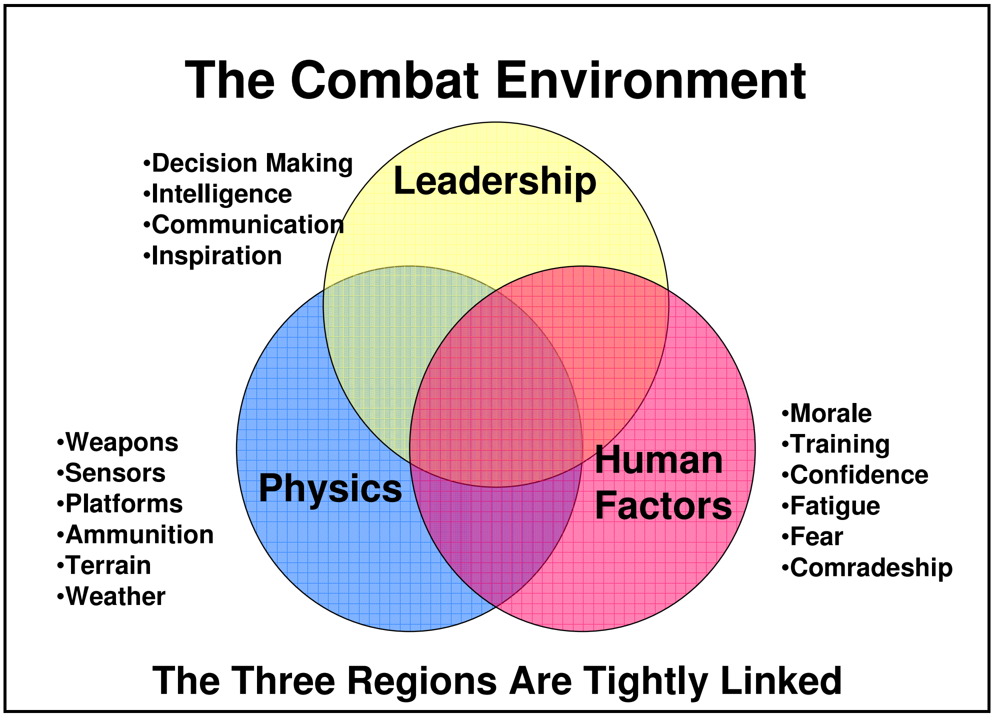


Figure 24: The Combat Environment [14]

Pythagoras was designed with three simple criteria in mind [14]:

- It must be simple enough to use that military officer with a nontechnical degree could learn it in no more than eight hours.
- In must incorporate fuzzy logic in some way.
- It must be *data farmable* (able to be executed in a batch mode on parallel processing computers).

The second criterion is what separates Pythagoras from the other tools discussed previously. Although EINSTein and MANA have the capability of assigning personality or action traits based on the environment, Pythagoras assigns one of three distributions around a personality trait. Depending on the settings for the agent rules, referred to as soft rules, an agent can have a firm/low individuality range, soft/medium individuality range, or a loose/high individuality range around the specified trait threshold. This allows the agent to have some flexibility in its reaction to existing conditions so that the agent can learn and or modify its behavior when necessary. Through this capability Pythagoras's intent is to model human behavior that is based on one or all of the five senses. This tool is currently also in use at the Naval Postgraduate School for the purposes of modeling military scenarios.

2.15 Chapter Summary

The purpose of this chapter was to give the reader a foundation and an overview of the new and exciting methods available that attempt to solve real world problems for designers and researchers that rely on some form of forecasting or prediction capability to make their decisions. The non-linear nature and elusive causal relationships not visible through traditional methods requires advancement in a DM's ability to predict the future. The basis of many of these methodologies is not necessarily the result of new concepts, but the application of new technology and innovative adaptation of theories in cognitive engineering that attempt to mimic HITL interaction and qualities. The military requires extensive HITL interaction, but also has some flexibility and need to incorporate automated processes to narrow the options available so that justifiable and transparent decisions can be made and specific courses of action can be pursued. Currently, gaps in information access and visualization capability of this data in a timely and efficient manner do not exist in the military logistician's realm. Facilitating data organization, access, and retrieval of relevant information with some of the methods described would aid in fulfilling the overarching goal outlined by military leaders who desire increased precision and quantifiable justification for course of action (COA) selections. Cognitive methods such as Case Based Reasoning, principles of *fuzzy logic*, and modeling and simulation have the propensity to give a user enhanced insight into a design environment.

3 TECHNICAL CHALLENGES, RESEARCH QUESTIONS, & HYPOTHESES

"The fact that the future is uncertain is no excuse for failing to make adequate preparations."

- USMC Operational Maneuver from the Sea

In the United States there are three general tests any invention, concept, or idea must go through to be accepted by the US Patent Office for a patent. Those three general tests are as follows [99]:

1. The item in question must be new or novel, in so much as the invention must be demonstrably different from publicly available ideas, inventions, or products (so-called "prior art"). This does not mean that every aspect of an invention must be novel. For example, new uses of known processes, machines, compositions of matter and materials are patentable. Incremental improvements on known processes also may be patentable.
2. The item must be useful. It must have some application or utility or be an improvement over existing products and/or techniques.
3. The item must be non-obvious, it cannot be obvious to a person of "ordinary skill" in the field; non-obviousness usually is demonstrated by showing that practicing the invention yields surprising, unexpected results.

Although the author is not getting a patent, the premises here are very similar to that of a thesis topic. In this section a re-familiarization of the problem is given and the technical challenges foreseen by the author are provided. With these challenges identified, research questions and ensuing hypotheses are presented to provide granularity to the goal of this research effort.

3.1 Re-familiarization of the Problem

The intent of this research is to develop a methodology that is capable of quickly providing information to military decision makers as to what architecture has the best probability of providing success no matter the operational environment. The continued downsizing of the United States military and its ever expanding involvement in diverse operational environments has placed all branches of the armed forces at a crossroads. How to continue to do their jobs effectively and efficiently? To accomplish this task the DOD along with select parts of the military have begun to integrate themselves into Joint Operational Forces that utilize the best practices, strengths, and other capabilities to find solutions to the requirements placed before them. The operations they are tasked to do vary from the low danger setting of humanitarian aid or observers to full fledged conflict with combatant enemy groups seen today in Afghanistan and Iraq. The bulk of the military's role holistically is to insure the safety of the United States, but the US's role on a global scale has the military functioning in many areas that are out of this scope or are minimally related to it. Faced with this operational environment, a major consideration is how does the military supply themselves when faced with such differing resource requirements from combat to humanitarian missions? The ability to not only have resources available capable of meeting these varying needs is a must, but predicting and or forecasting the resource needs has become a top priority for the logistician in the U.S. armed forces.

The need for this capability based predictive analysis is called for by various government and military agencies. A methodology to support this desire has yet to be created, in part because of the large volume of entities and items that would need to be tracked. Recent and continued development of techniques in capturing real world information, practices, and ill defined requirements now make revisiting an endeavor like this practical and possible. A simulation environment to model and gather data that depicts the top level and primary resources used in an operation would be useful. For that reason, a software package called MANA was selected and will be further explained in Chapter 5. The operational environment is one that involves experts in the form of logisticians, other human decision makers, and a highly non linear environment when attempting to predict or forecast resource needs. These characteristics fit very well in the class of other real world problems, which have been investigated, modeled, and proven to have better predictive results than traditional methods when analyzed with non linear methods. Examples of these applications can be seen in Appendix E.

- The goal of this research and thesis is to develop a holistic capability based design methodology that allows for the development of a system capable of predicting and forecasting critical needs for joint forces in dynamic and fluid operational environments.

3.2 Paradigm Shift in Military Logistics and Architecture Solutions

For many years the military has planned, war-gamed, and conducted a variety of functions based on high level assessments and tactics from senior military and civilian leaders and analysts. This approach is often referred to as the top down approach. This process entails numerous complex and highly integrated activities ranging from planning troop movements to analyzing a variety and often times conflicting intelligence reports to generate a plan or course of action. The way the military was expected to operate in the past was in the form of a large scale force with multiple and supporting operations generally occurring in a single region. As has been alluded to in Chapter 1 these types of operations seem less likely in today's military environment. The operational scope for military units now embodies relatively small, fragmented, and multiregional conflicts. This realization has necessitated a change in the mentality of how to best prepare, plan and conduct these missions. Senior military leaders realize they still have the responsibility of providing the overall commander's intent, but in order to successfully coordinate, operate, and support the units operating in evolving and dynamic environments the bulk of planning and requirements generation must come from the bottom up. This approach allows operational and small unit level commanders to identify and request resources unique and specific to their operational needs. As this information is processed up, resources and planning for efficiency of resources and forces can be accomplished at the top level.

3.2.1 The Human in the Loop (HITL) Factor

Recently industry, the military, and a variety of businesses have sought ways to promote efficiency in the things they do. One such approach is to incorporate a variety of technological advancements that eliminate the human that once was required in the process. Many applications of enabling technologies that increase precision, capacity, or flow rate have been extremely successful in this endeavor, especially where rigor and well defined constraints or requirements are prevalent. Unfortunately, there are many aspects in design and problem resolution that lack concrete definition simply because information is not available or ambiguity still remains in the requirements.

Robustness in solutions has become the main stay in today's design and solutions environment. Researchers, scientist, and industry continually look for ways to apply technologies and concepts that can model the way human's think and process information. In Chapter 2 and Appendix D the concepts of artificial neural networks and fuzzy logic are discussed that represent this pursuit. Significant strides in these and related fields have resulted in advancements in many areas ranging from manufacturing to decision support and solutions implementation. For the military, the latter is of importance to them primarily because of the volumes of data available to DM when trying to determine a viable course of action. As was discussed, ANN while helpful, still have only scratched the surface in their ability to match the processing and computing power of the human brain. ANN's also require significant training to be applied to a problem area. For the military the information that would serve as the training data has not and sometimes can not be quantified. Another dilemma the military faces is that the operational environment they operate in and face for the foreseeable future is constantly in flux. Fuzzy logic and ANN work off of a rule based system, but as this rule based system grows efficiency diminishes and rules that aid in determining a solution start to incur conflicting consequents. These reasons outline not only the deficiencies in these methods and techniques, but also give researchers direction in what problems need to be solved for the future application of these concepts.

The HITL alone is not efficient – that has readily been proven in many instances, but neither is an artificial representation of a human where critical decisions that have global and far reaching effects need to be made. What is viable is a combination of the two. Information management if done properly can help military planners and decision makers to significantly reduce their decision processes and reaction time when evaluating and planning operations in time sensitive environments. The HITL will likely remain a critical factor for a long time to come if not for the reasons identified above, but also because it is highly unlikely that one will be able to completely translate human intuition into the logic that defines computers or artificial human processes.

3.2.2 Top-down Planning

This type of planning starts at the senior leadership level and tends to focus on the strategy necessary to meet an overarching goal or intent. Characteristics of this type of planning process consist of the following general attributes, besides what has already been stated [133]:

- Goals are determined early on in the process
- Inflexibility as processes are enacted
- Lack of participation from the operational (lower) level
- Long term goal oriented

Top-down planning affords a wide range of activities for senior level leaders to attempt to implement a global or high level vision for an organization. While helpful in providing guidance, its sole practice often times alienates those at the operational level, especially if senior level decision makers do not have an adequate understanding of the operational level needs. Because of this, the strategic vision will likely fail to be realized or implemented and can cause an initiative to be stopped in its tracks or before it can even start to work [140].

For the military this type of planning is primarily used during large operational conflicts where significant coordination and timed operations necessitate an operational flow to be successful. An example of this type operation is the D-day invasion during World War II. Several operations consisting of amphibious operations off the coast of France in concert with airborne operations inland were meant to disrupt the German Army with a two pronged effort, on land and from the sea. As history has recorded, many lives were lost that fateful day by soldiers who were following orders and implementing the plan created by senior leaders. Some would argue whether the operations were a success or a failure due to the loss of life, but what is clear is that this operation focused on a clear and absolute threat and is generally considered the turning point in the war in which the Allies began to drive back the advancing German Army. Today the operational environment for the military is fragmented, asymmetric, and ambiguous. The enemy, primarily terrorist, are harder to classify and detect. They operate using guerilla warfare and are funded through a variety of mediums that allow them to operate as extremely small organizations with differing goals and tactics. For that reason an approach that aims to customize a preventative and reactionary force commensurate with the operational environment and unique requirements for the area is necessary.

3.2.3 Bottom-up Planning

This process tends to focus on the tactics or ways to accomplish a goal. Plans are developed at the lowest levels and are then passed on to each next higher level. It then reaches senior leaders for approval. This process primarily seeks to garner buy in from the lower level participant by involving those who will be doing a job from the beginning of the process. This type of planning process is often characterized by the following [133] and [140]:

- Flexibility
- Instills teamwork
- Accountability a all levels
- Lacks long-term vision

Bottom-up planning allows the most up to date practices and information to flow up from the people actually doing the job and provides justification for resources or as to why a specific course of action is being pursued. Senior leaders can then take these plans and review, modify, and or approve them. Failure to manage these plans properly can result in local optimization in areas that do not capitalize in efficiencies of resource allocation nor converge on a top level intent.

Recently in the military, this type of planning has been advocated as the best way to meet the operational environment discussed in Chapter 1. No longer are the large operations or well defined adversaries prevalent or identifiable. Operations in one part of a region may significantly differ from those just a few hundred miles away. Senior leaders have realized that operational and low level commanders require some flexibility and autonomy to be successful in defeating their regional adversaries. In order to do this, their planning and requirements generation for logistics and equipment must be met quickly and efficiently.

3.2.4 Conclusion: Which is better?

Top-down planning allows for an overarching strategy to be outlined and pursued. Bottom-up planning affords the implementation of specific and relevant tactics to be applied at the operational level. As with many things in a complex environment, the answer on which process is the best is, it depends. The operational landscape facing the military and many entities today demands and almost requires robustness in solutions and pursued course of actions. To accomplish this, organizations must now find a suitable balance between top-down and bottom-up planning.

For the military this means providing operational level commanders general guidance in the goals being sought and allowing these commanders a lot more flexibility in how they accomplish their missions and providing the support, logistically and other materiel, needed for particular area of operation. Achieving this balance can be difficult since ultimately senior level commanders are held accountable for the successes and failures of those under their command. This type of climate tends to lead leaders toward strong oversight and often times micromanagement. Fortunately the growing belief and recommendation by many senior leaders in the military is that delegation and relinquishing of control of planning at the operational level must be pursued and perfected if the military is to meet the dynamic and emerging threats that the military is being asked with increased frequency to contend with [57] and [97].

This change in strategy should help to enhance the military's pursuit in facilitating information flow up from the bottom to the top and down from the top to the operational level. While a good evolution of the military command and information structure, there exist other challenges that also need resolution. One such area is the modeling and simulation community that is tasked to model, evaluate, and analyze the variety of military operational environments that each has their own set of unique and complex issues. Some of the challenges will be discussed next.

3.3 Technical Challenges

Computer simulation has seen and explosive emergence in the area of complex non linear systems. While ongoing research and application has yielded some interesting and helpful results in many applications, it is still not a perfect or exact science.

Arguably many people consider modeling and simulation (M&S) of these systems more of an art that a science and it presents several technical challenges:

- C1: Modeling military systems is not only complex because of the number of variable involved, but these simulations often times have to be simplified smartly so that executing explorations of the design space are efficient and timely.
- C2: The model built must accurately represent the environment being explored. Getting agreement on the fidelity of a model among participants, designers, and experts can be very difficult.
- C3: Incorporating cognitive processes into a model, although not new in concept, where a human in the loop is necessary adds increased complexity to an already complex problem.
- C4: Providing transparency of the solution presented by a model is challenging. As the capabilities of systems increase, their complexities rise, thus reducing the transparency to all but the designer. The novice user must be able to have a traceable path to the solution.
- C5: Sufficiency of the "answer" and is it useful should be at the questions at the forefront. Determining this for new simulation systems where no empirical data is available will be difficult. Initial stages may require extensive verification, validation, and accreditation to be accepted by the user community.

Additionally to the specific challenges outlined above, some of the other general challenges that the research community, hence the author, face are:

- C6: Often times there are no set guidelines or clear best practices identified. This is true even in areas where success has been achieved with particular techniques because each application tends to be unique and determined heuristically.

- C7: Optimization once used to be the goal of many research efforts, but this trend is diminishing as not only robustness is being sought, but the desire to be able to react quicker and dynamically change to meet varying needs as they arise is becoming the norm.

- C8: Determining the right measure of effectiveness to be observing or tracking is difficult. Because the desires and needs of all the participants vary, especially depending on the scenario environment, flexibility / fidelity of a model must be balanced accordingly.

- C9: Usability and cost are always factors in implementation of a process. Accounting for this in the beginning is essential for any research to applicable for "real world" implementation.

These technical challenges help to foster questions the author has on how to adequately approach the aforementioned research area and will be covered in the next section. Subsequently a methodology developed and reviewed in Chapter 4 will explain how these challenges can be addressed in the military logistics supply arena.

3.4 Research Questions (RQ) & Hypotheses

"As complexity rises, precise statements lose meaning and meaningful statements lose precision."

- Lofti Zadeh, Father of Fuzzy Logic

The technical challenges expressed above were the result of a detailed definition of the problem being explored by the author for military decision support systems in the specific area of logistics. A research question (RQ) is a formal inquiry that follows one or more observations after a detailed exploration and understanding of a design problem or environment. A hypothesis is a proposed solution to one or more research questions that can be tested via experiments. These experiments are usually in the form of broad research tasks, which often have many sub-components that can be used to guide a methodology aimed at providing a solution or shedding further insight to the problem. The formalized process may either confirm the hypothesis/hypotheses or lead to other observations. The following discussion expounds upon a set of research questions from the authors investigation into decision support systems for military DM's.

In order to identify the logistic framework best suited supporting military units, a narrowing of the scope of the over all problem must be performed to identify applicable mission scenario variables and requirements. Given those parameters, analysis can be conducted via decision support systems tools to allow a decision maker to make supported decisions. Unfortunately those parameters are uncertain in part because intelligence and surveillance data can vary significantly depending on the source and force size for missions can dynamically change based on other factors beyond a DM's control.

3.4.1 Hypothesis 1

RQ 1: How can a process be created that enables rapid extrapolation of viable courses of action for determining military unit level logistical support architectures?

- Data management is the foundation for utilizing information, especially data that is rapidly arriving from multiple sources. Leveraging hardware and software applications can help if applied efficiently and uniformly across a communication network. Time is critical in digesting and utilizing this information so easy to understand, rapid, proper fidelity DSS tools will be needed. Knowledge Engineering and Case Based Reasoning Techniques, discussed in Section 2.3 and Section 2.3.3

respectively, have been applied in a variety of fields for assessment and identification of likely prognosis based on selected initial conditions.

- Defining, building, and executing M&S software to visualize a specific well defined scenario may be feasible, but in dynamic and unstable environment a DM would be hard pressed to come up with a justifiable solution in the time allotted. This environment can be seen in Figure 3. To alleviate this issue, meta-models that represent the design space can be substituted to increase exploration of the design space. For this non linear space artificial neural networks, discussed in Section 2.8 and 2.11 should be used.
- Organization of data in an easy to process manner has allowed cases to be entered and evaluated in areas such as diagnosis of breast cancer. By allowing new cases to be added to existing information in a database of known inputs and outcomes, the likely resulting accurate diagnosis of cancer identification has improved dramatically. It has also allowed doctors to mitigate diagnosis errors and increase their response times to patients who are anxiously awaiting their results.

RQ 2: How can the critical variables be identified that affect a specific architecture selection when the operational environment is dynamically changing?

- Dynamically changing environments will have an impact on COA selection and other decision making activities. Understanding the ability and capability of soldiers, commanders, and other decision makers involved will help ease the difficulty in identifying the variables associated with the dynamic military logistics environments. Variables can be identified that for the most part have an influence on the outcome through careful dissection of recent problems. Adjustments and assessments of these variables may be needed from time to time as the operational landscape changes or adversaries modify their tactic and strategies.

3.4.1.1 Hypothesis Statement

> *Any method formulated to provide a decision support system for determining military logistics architectures should implement case based reasoning and fuzzy logic methods to provide quicker access to relevant information that will reduce the decision making process.*

The viability and feasibility of any logistics support structure for a small unit force can be determined by establishing a base unit from which all others are built. HITL participants, soldiers in this case, are innovative, driven, and resilient under harsh conditions when required. Because there is limit to this capability, a determination of potential outcomes across a range of probable inputs should be conducted to allow a DM to justify the need for particular resources based on operational scenario characteristics.

3.4.2 Hypothesis 2

RQ 3: How can the quantitative aspects of a scenario environment be accounted for and allow for the incorporation of expert knowledge?

- HITL participation is a requirement in this environment and just cannot be eliminated. The military is an organization whose people are integral cogs of the decision making process. Their personal assessments may vary depending on their experience, but if they know what variables are important they can generally process the information provided to arrive at an acceptable plan of action. Communication and an understanding between the logistician (DM) and commander on the ground are integral to creating a plan that is best suited for the situation.

- Modeling and simulation has provided a wealth of capabilities in prediction and forecasting analysis. Agent based techniques discussed in Section 2.6 and 2.7 discuss general application of methods that have allowed realistic representations of environments and actors in that environment. The ability to model agents, their interactions, and capabilities continues to increase. Current state of the art in these techniques, especially for military applications was presented in Section 2.14. The software application, MANA discussed in section 2.14.4 will be utilized because of its ability to satisfactorily model the notional proof of concept being investigated.

RQ 4: How can initial conditions and outcomes of past COA's be integrated into the DSS to facilitate better predictions of viable architecture selections?

- Once enough data is acquired or simulated through vetted modeling and simulation tools, probabilistic distributions can be associated with the inputs or design variables for the logistics support structure. Available resource allocation will be critical in determining capabilities, but the ability to think out of the box in terms of capabilities and alternative COA is why the HITL must remain and be a part of the decision making process. Providing a way for the HITL's expert knowledge to be incorporated into the DSS that helps determine the logistics architecture should be incorporated. The primary M&S tool, MANA, contains reports that will be utilized to extract the measures of effectiveness (MOE) and resupply data necessary to build a suitable case based database.

3.4.2.2 Hypothesis Statement

> *Historical or simulation data that provide low level characteristics can be used to quickly filter solutions for the decision maker through data based management techniques.*

Cognitive science is an area of research that will likely have no end in research opportunities. Modeling the human mind is a difficult and complex problem that is more of an art than a science. Adaptations made possible by simplified computer models and techniques have helped DM gain an understanding in a variety of ways to find solutions to problems and provide useful insight to highly nonlinear interactive environments.

3.4.3 Hypothesis 3

RQ 5: How can qualitative aspects of a scenario environment be accounted for and allow for the incorporation of expert knowledge?

- People describe things in the form of words or linguistically. Applying linguistics to problem resolution properly allows for a level of imprecision that can result in outcomes that mirror real world events. Applying membership functions across a range of acceptable quantitative answers has proven very effective to this end in a variety of applications. Section 2.10 and Section 2.11 discuss the underlying principles of *fuzzy logic* and Section 2.13.3 provides some state of the art applications of these concepts that allow cognitive reasoning techniques like soft computing and AR, discussed in Section 2.3.1 and Section2.3.2 respectively, to be used to facilitate the reduction of the decision making process cycle time.

RQ 6: How can the uncertainty associated with current information be incorporated prior to and during an operation to justify a COA during the decision making processes?

- As information becomes clearer and a situation develops uncertainty in information generally diminishes and confidence increases. Allowing this aspect of the scenario environment to be accounted for via a Graphical User Interface that allows the user to compare current data to case based data at any temporal point would be useful.

- Accounting for uncertainty in random systems has gotten easier with statistical techniques and technology advancements in recent years. Statistical sampling has served well to represent and aid in solving complex problems that often have no exact numerical solution. Reducing uncertainty is challenging when linguistics are involved. Gaining consensus among decision makers prior to a system or methodology implementation can aid in limiting the sensitivity that controllable factors have on the outcome.

3.4.3.3 Hypothesis Statement

> *Fuzzy Logic principles ought to be used to account for human in the loop, i.e. the decision maker's assessments of the situation when utilizing a DSS to determine a course of action.*

Two schools of thought exist when it concerns the human in the loop. The first is that the HITL is inefficient and should be eliminated at all costs. The second contends that the HITL is a necessary component of the decision making process and can not be eliminated. The likely answer to which contention is the right answer is that it depends on the situation. There are several processes or activities that are simplistic, repetitive, or require exacting precision that a human influence degrades the outcome. Yet, there are processes where computer technology is incapable of adequately, safely, and reliably making or executing the right decisions without some form of human direction. Striking a balance between these two schools of thought is best left up to and must be determined by the designers, researchers, and decision makers involved.

3.4.4 Hypothesis 4

RQ 7: How is robustness incorporated into architecture selection to mitigate variations in the uncertainty of the operational environment?

- It is important to realize that as the HITL is an integral part of the decision making process, case based solutions will likely be the best starting points and not the exact or optimized solutions. Monte Carlo simulations, discussed in Section 2.9, allow for variability in the input variables to be tracked by adding distribution types to any of the point values. This technique is useful because it can be executed rather quickly and there exists excellent software packages that allow efficient and beneficial results analysis and visualization for the user.
- Several techniques have been developed that allow for environment with many design variables to systematically be vector-ized and normalized for comparison purposes. Section 2.4 gives an overview of three such methods. Allowing weighted factors to have an effect may or may not be necessary, but integration of this capability should be beneficial in accounting for the variability associated with the interpretation of human linguistics.

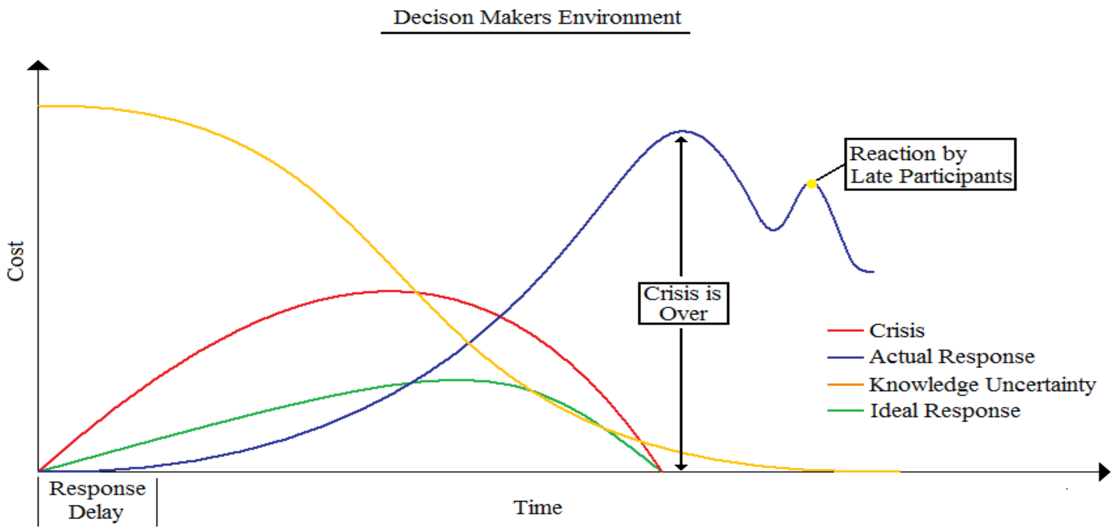
3.4.4.4 Hypothesis Statement

> *Optimization techniques should be used to account for uncertainty in information and aid in providing transparency and justification for a COA taken.*

Today's design environment has evolved into a world where point solutions are no longer relevant. Solutions need to be robust, flexible, and capable of meeting the needs of a varied and unique family of problems. As a result a family of solutions needs to be sought. Optimization affords a glimpse at point solutions, but repetitive optimization while perturbing the inputs should allow designers and decision makers to gain insight to emerging trends that can be applied as a baseline for the robust solution being sought. Optimization still has its uses and is a viable way to narrow down the sea of information and data inherent in today's design problems.

4 METHODOLOGY

The current state of the art in forecasting methodologies has come a long way from assuming stable demand and no accounting for "spikes" in demand in the operational environment was the norm. While modeling and predicting reality is getting better and better every year, there still is much to be done in providing methods that are capable of quickly and efficiently providing information to DMs in volatile and dynamically changing environments. These design spaces require accurate forecasting to determine the appropriate architecture, identify critical nodes, and efficient processes that facilitate robustness and increase the probability of success. These methods must now also allow or incorporate the ability to account for variability and uncertainty that generally originate from HITL interactions and a variety of other noise factors. To that end, a methodology called the Adaptive Technique for Logistics Architecture Selection (ATLAS) has been created. For this research ATLAS will be applied to the complex DOD logistics system in support of the near future concepts, Seabasing and Distributed Operations. This environment is indicative of the figure below from Section 2.2.



Although there are many metrics one might be interested in tracking or assessing, one could argue that ultimately everything comes down to some form of *cost* associated with a delay in a decision as time progresses. For the military this cost is categorized in monetary terms, national and international security, and more importantly soldiers lives.

In an attempt to formulate a method capable of providing insight to DMs via emergent behaviors concerning the logistics architecture and selected COA's, a thorough investigation into the critical components of the problem as well as potential methods capable of modeling the environment was conducted. This investigation was performed for the purposes of identifying what is appropriate to be modeled as well as how to model the environment. The morphological matrix, shown in Table 4, is a partial representation of the various options available to a researcher to begin modeling and analyzing the

Table 4: Morphological Matrix for Assessing Concept Alternatives

Concepts	Available Methods or Tools				
Requirement Definition	Customer	Historical Trends	Forecasting Analysis	Stochastic Processes	
Operational Environment	Static	Dynamic			
Modeling Uncertainty	Probability	Fuzzy Logic	Classical Sets	Info Gap	Dempster-Schaeffer
Facilitate Forecasting	Delphi Technique	Case Based Reasoning	Soft Computing	Approximate reasoning	Adaptable networks
Design Space Exploration	Full Factorial	Space filling Sphere pack	Random-Uniform	Latin Hyper Cube	Monte Carlo
Determine Problem Representation	Nonlinear	Linear			
Determine High Value Nodes	Pareto	Customer Input	Prediction Profiler	ANOVA	
Surrogate Modeling	Linear Methods	RSE	Neural Networks	Markov Process	Fuzzy – NN
Provide Prediction Capability	Markov Process	Case Based Reasoning	Wiener Filters	Bayesian Networks	Regression Analysis
Scenario Generation	Existing tools	Custom tool	MIP		
Decision Making Capacity	Human in the loop (HITL)	Artificial Intelligence	Adaptable system	Soft Computing	Combination HITL & AI
Use of Real Time Sensory Data	User level	Operational level	Top level		
Simulation and Visualization	ARENA	Pythagoras	MANA	Custom	MatLab

necessary information for a DM to select a support architecture and COA for this environment type. This matrix and the resulting assessment is focused toward the problem at hand, the architecture selection and insight into the logistical distribution network for a military force faced with a broad spectrum of responsibilities. The matrix is not an all inclusive representation of alternatives available, but shows there are at least 1.8 million possible methodologies that could be used to aid the DM in forecasting the logistical needs and determining the systems architectures.

The morphological matrix in Table 4 addresses C6, where the author contends there exists no one "right" way to solve a problem. Instead a decision must be made based on the design space or operational environment as to what tools, techniques, or custom processes need to be used or created so that a problem can be solved. A down select was conducted based on the author's assessment to guide him to a unique solutions process capable of defining, representing, and analyzing the military logistics analysis problem described in Chapter 1.

The areas highlighted in yellow helped the author narrow down the potential methods or tools available to formulate a direction in the research into joint logistics forecasting. This was accomplished by examining the problem and the needs to model the problem. For instance the first item "Requirements Definition" has four potential options. Because the current military environment is being shaped everyday it is important to have customer input into the process. Although determining the most robust logistics architecture is a dynamic and evolving problem, historical trends (recent and old) can be utilized model "real world" future expectations. The other two options could have also been used, but were eliminated by the author. This same approach was taken for all of the concepts in the matrix to down select which tools or approaches the author felt were applicable to the DOD problem. Although a scoping of the tools and methods was performed this still represented 15,552 potential combinations as indicated by just the yellow shaded cells in Table 4. Continued investigation into these options approaches was conducted and a further elimination of those items in yellow was performed; indicated by the diagonal thatching of the shaded cells. The result is a more manageable 16 approaches and is based on the assumption of a combinatorial application of things like Pareto analysis and a prediction profiler or the combination of using MANA for M&S in conjunction with MatLab for visualization would be conducted.

In order to provide a logical selection and management of the information available to the DM, the creation of a logical process that begins with problem definition to solution needs to be created. The aim should be to capture pertinent information in such a way that it narrows down the volume of superfluous data and allows a DM's to visualize relevant performance measures and ultimately facilitates the selection of COAs. The steps a designer in this process, which defines the ATLAS method, would take are outlined below:

A	Step 1: Define the Problem
	Step 2: Identify Systems Architecture and Major Components: Experiments
	Step 3: Create Modeling and Simulation (M&S) Environment
	Step 4: Verify M&S Environment
B	Step 5: Create / Update a Case Based Database
	Step 6: Identify and Incorporate Uncertainty in Knowledge or Information
	Step 7: Create Surrogate Model
C	Step 8: Visualize and Explore the Design Space
	Step 9: COA for Architecture Selection or Iterate Back to Step 5 or Step 1 if Necessary

The methodology above is a very broad and representative of any problem decomposition and subsequent processes aimed at determining a solution method in the sense that it is applicable to a variety of problems. While the goal of the steps above is to have a method that is applicable to more that just the military logistics problem being investigated, the ATLAS method must be further defined to be applied to this class of problem. This is further outlined in the following more specific steps:

A	Step 1: Categorize the problem in terms of military operations (i.e. combat, humanitarian aid, counter terrorism, etc.
	Step 2: Identify system level architecture and Components in terms of personnel, equipment, and expected enemy characteristics.
	Step 3: Create a Modeling and Simulation (M&S) Environment that represents the operational (low) level characteristics in an appropriate military M&S application.
	Step 4: Verify that the M&S environment is representing realistic and or expected results.
B	Step 5: Create / Update a Database of specific inputs and experienced or simulated outputs that can be accessed via a user's query using Case Based Reasoning.
	Step 6: Identify and Incorporate Uncertainty in Knowledge or Information via Fuzzy Logic and the incorporation of Fuzzy Membership Functions.
	Step 7: Create Surrogate Model that is representative of the environment. For military operations this environment is asymmetric, dynamic, and non linear. ANN is best suited for this environment type.
C	Step 8: Use the Case Based Database to Visualize and Explore the new Design Space as compared to those in the database.
	Step 9: Select a COA for Architecture Selection and forecast the need for critical logistics materiel or Iterate Back to Step 5 or Step 1 if Necessary.

The methodology will be applied to aid the military DMs in determining what the logistical needs are and the force architectures that are capable and robust enough to meet the needs of changing and dynamic missions envisioned for the future force described herein. The ATLAS method encompasses a range of techniques that should have explanatory properties capable of imparting knowledge to the military DM and increase their awareness of the problem domain. Pictorially represented in Figure 25 is a flow

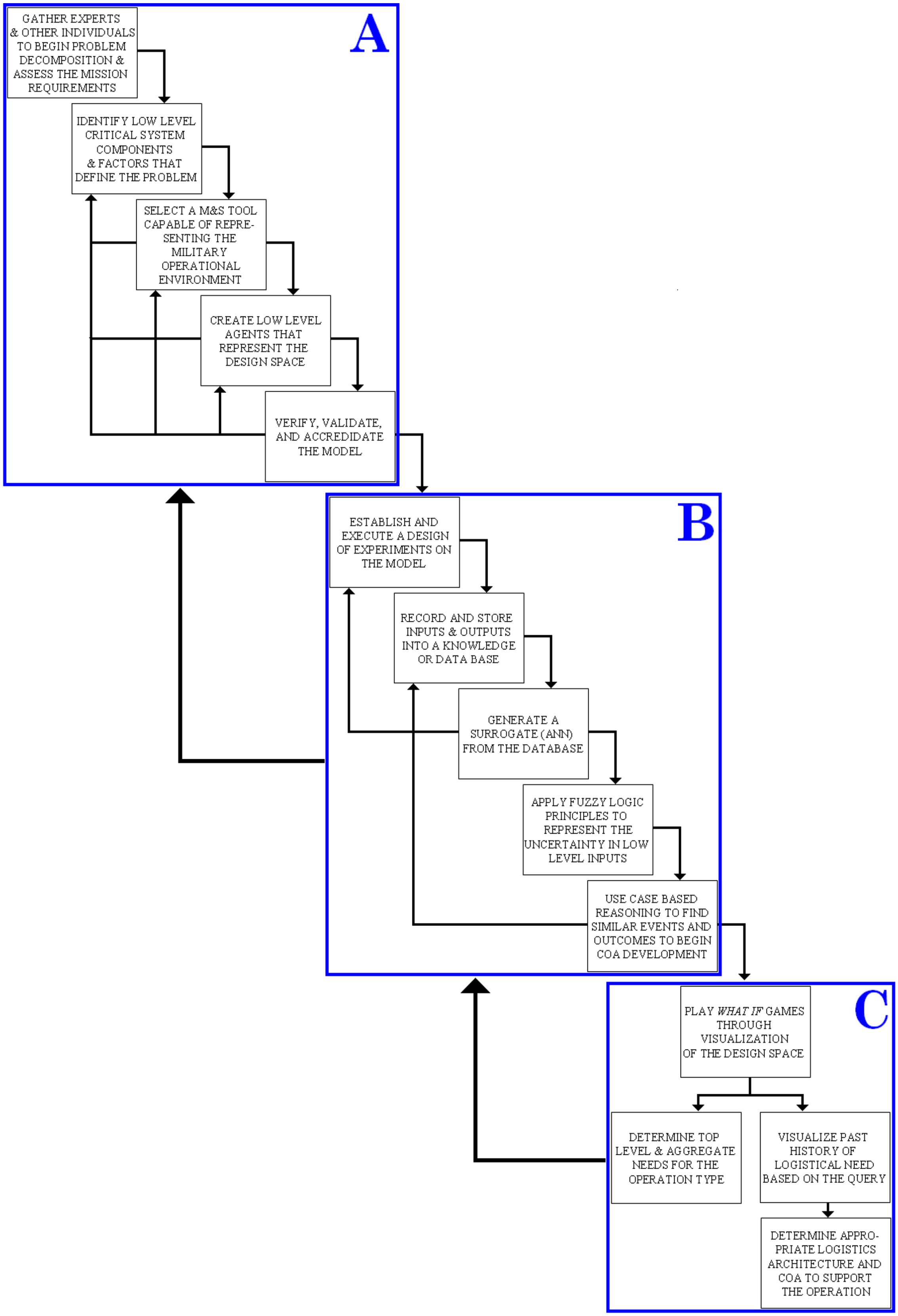


Figure 25: Flow Chart of the ATLAS Methodology

diagram of the specific steps that will be employed by the author in his investigation of logistic determination for a small military unit under the Seabasing and Distributed Operation framework.

The first process block of ATLAS method, A, is to develop the modeling and simulation environment. This is where the designer must assess the problem posed. This encompasses getting the DM, experts, and or others who have a stake in the outcome involved in the process as soon as possible to determine measures of performance, measures of effectiveness, and reach an agreement on a suitable framework for modeling the scenario environment. The fidelity of the modeling environment should be dictated

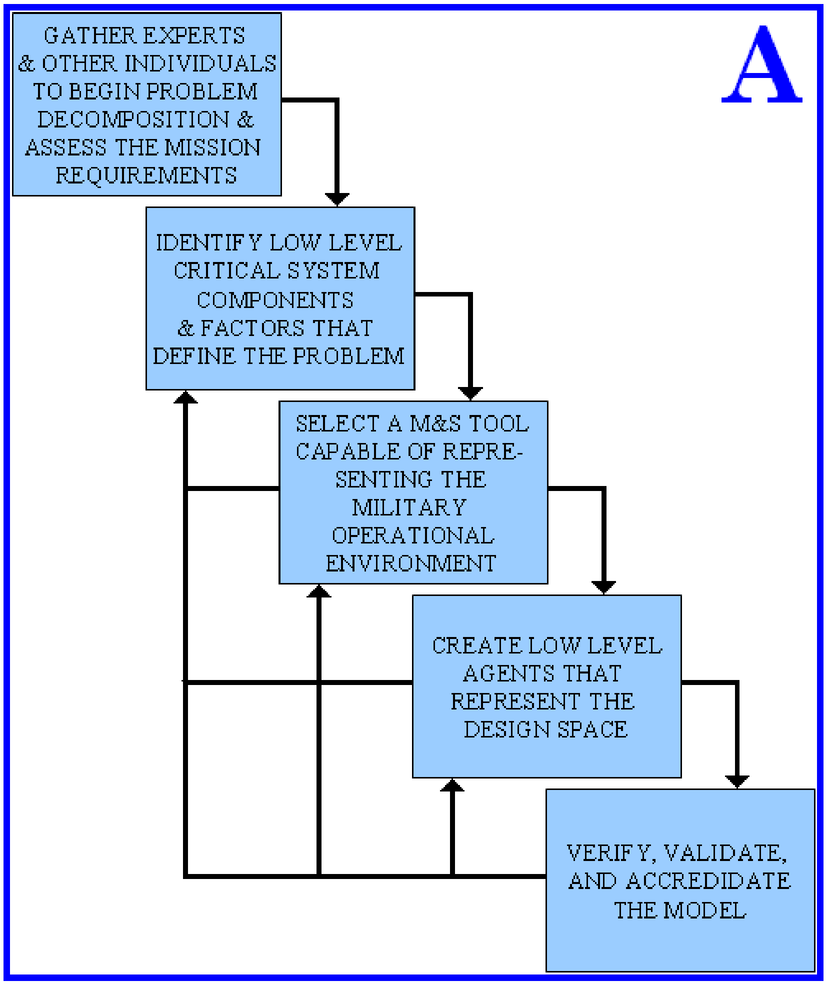


Figure 26: Develop / Modify the M&S Environment

by predefined critical nodes and relationships thought to exist or deemed necessary when constructing the model. This process can be iterated upon if necessary as indicated by feedback from the third process component labeled C in Figure 25. By performing these steps in the ATLAS method, theoretically a designer should be able to overcome C8 so that the MOE's, the model, and the results obtained from subsequent analysis are usable to anyone involved in the design and COA selection process.

The second process block, B, of ATLAS develops or modifies the model from process A. The dynamic environment of military operational space is complex. Helping to reduce that complexity is one of the key components of ATLAS. This is accomplished

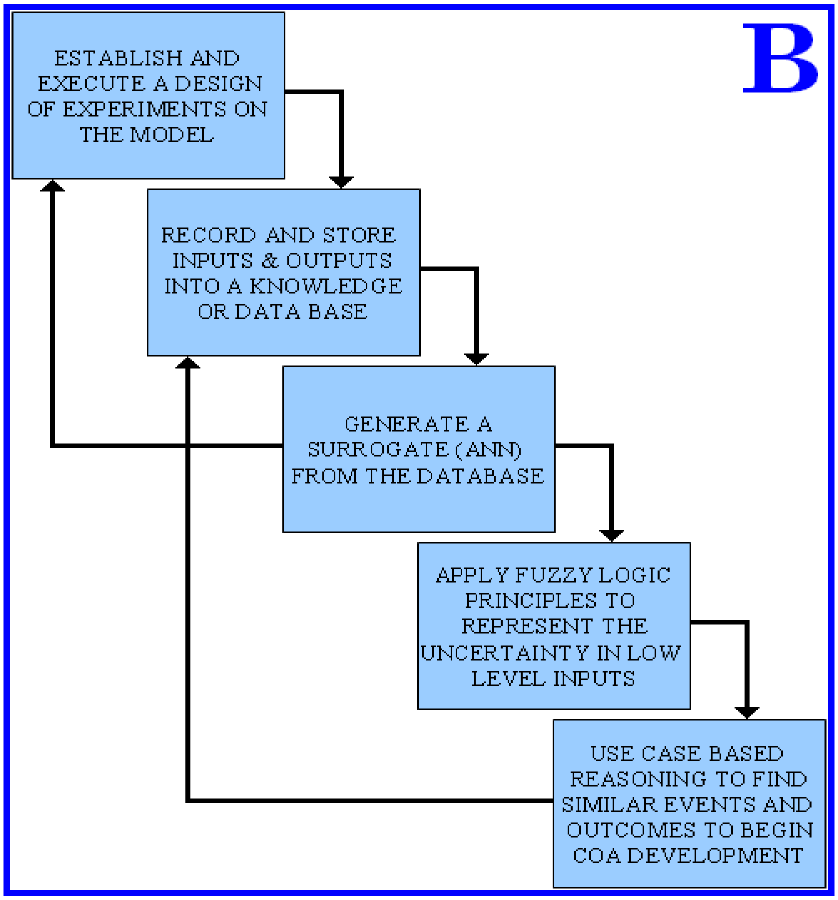


Figure 27: Create / Modify the Surrogate Model

by providing a way of gathering the knowledge, experiences, and information that can be effectively applied to the design space into one central knowledge base. Designs of Experiments should be instituted to execute the model over a range of input variable settings to populate the knowledge database and to aid in generating a surrogate model of the operational environment being explored. The knowledge database is meant to aid in the selection of appropriate COAs based on current force architecture and situational information through queries by a logistics DM or commander. As with any evolving and real world problem, there exists some level of uncertainty in the information available. Accounting for this uncertainty will add to the increased fidelity of the surrogate that is created in this component of the method.

The third component, C, in ATLAS is the analysis block. This is where correlations are observed and data is scrutinized. Through visualization software, *what-if*

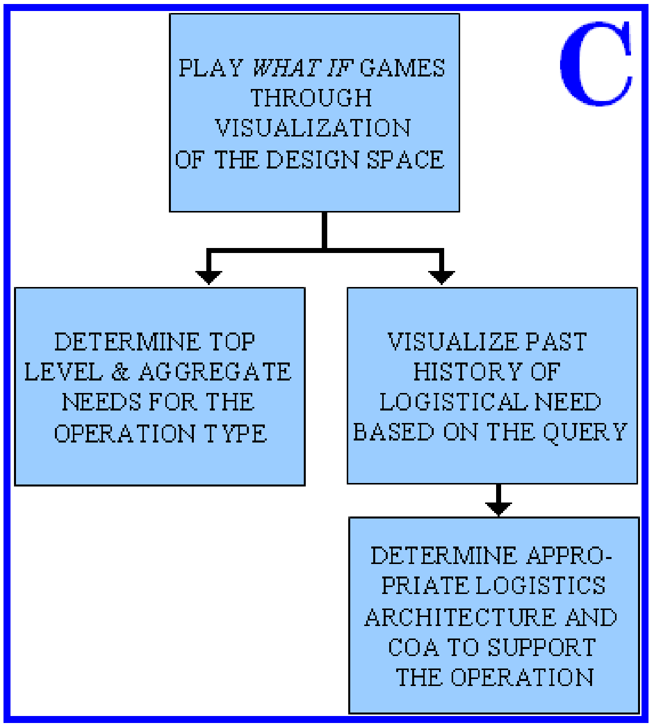


Figure 28: Perform Analysis and Select a Course of Action

investigation can be performed and DM's can potentially see the impact of various COA. COAs are executed and then feedback is instituted to measure how "good" the DM's choices were. If the results are acceptable and a COA is pursued the process stops and the scenario information can be incorporated into the case based database for future use. If no COA is determined, the process will iterate back to component B for reevaluation. If the result or no suitable COA can be determined the process can go back to the initial development of the M&S environment, component A, to better define and or represent the design space. This process can continue until the human in the loop DM is satisfied with the forecasted architecture needs, logistics requirements, and predicted outcomes on the MOEs or MOPs. The icon proposed for the ATLAS method is shown in Figure 29.

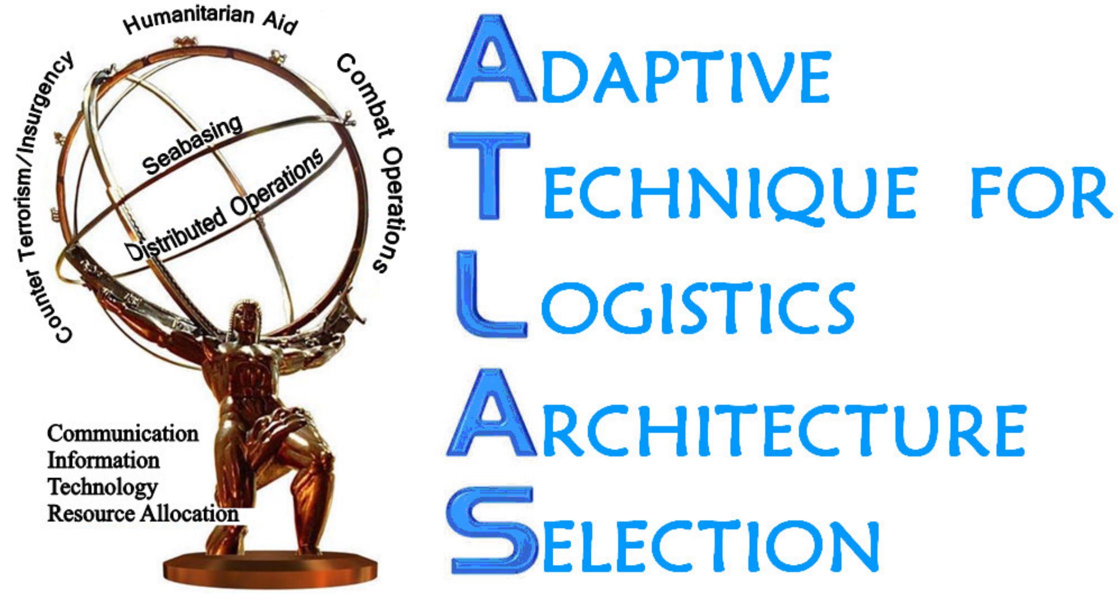


Figure 29: ATLAS Icon

The ATLAS method is a decision support system/tool that is meant to aid in the DMs selection of system architecture via a generation of a requirements list to support and sustain operations across the broad spectrum of environments. For the military, these spaces primarily fall under counter terrorism, humanitarian aid, and combat operations. At its core the DOD and armed services are pursuing the concepts of Seabasing and Distributed Operations to meet the robust architecture and flexibility needs in the littorals. This will be done by leveraging continuing advances in communications capabilities, information gathering, implementation of new technologies, and efficient resource allocation.

4.1 Methodology Defined

The ATLAS method has been developed not only as a way to attempt to solve the military logistics and architecture problem being investigated, but also as a way to address and overcome some of the technical challenges outlined in Section 3.3. These challenges are ones faced by designers, engineers, and analysts in a variety of fields on a continual basis. Evolution of techniques, software, and other tools are implemented through the ATLAS method to meet these challenges. Ultimately, application of these solution types will have to be applied as more and more challenges are indentified and to those that are as of yet an enigma to the design community.

4.1.1 Step 1: Define the Problem

As with any solution process, the first step should be to understand the problem as much as possible before venturing down a path that attempts to find an answer. Table 4 shows there are several paths available to those attempting to find a solution process. The volume of possibilities represented here reinforces the need for this critical step to create a solid foundation to the created solution process. Defining a problem can occur in a variety of ways. The options available to perform this step will vary depending on the situation, but often times problem definition occurs through observation, problem reports, and or some form of analysis.

Observation is the simplest form of problem definition that gives a person visual cues to how a process currently works and how it may be failing or lacking to some degree. This form while quick and easy can often leads an observer to only see the surface of the problem and can fail to identify the root cause of a problem. Reports or information on problems are often times collected by groups or entities and can be aggregated to show some trend or apparent cause to issues facing the problem solver. This technique tends to add some reliability to identification process because hard data substantiates the problem identification process. While this is true, this method can be skewed as what is one problem to one person may not necessarily be an issue to another, thus causing some ambiguity as to what are the driving factors attributed to the problem being investigated. The last method of analysis tries to incorporate a balance of all three methods with emphasis on delving as deep as the problem solver wants to in trying to identify causal factors. Applying this process usually results in a deeper understanding of the problem space on a holistic scale and can be very beneficial to quickly finding a solution. The caveat to this method is that it is often time consuming and expensive to perform. Depending on the situation, this method might also be unnecessary if the causal factors are easily indentified by another method that can adequately define the problem.

4.1.2 Step 2: Identify Systems Architecture and Major Components

With the first step accomplished, the next step is to identify the environment and components relevant to the system or systems being analyzed. This process should consist of a multidisciplinary or conglomerate of participants. Today's complex problems often have a variety of participants, large and small, that are in one way or another are a stakeholder who has an interest in the problem being investigated. Inclusion of the pertinent participants goes hand in hand with Step 1 and also extends to this step to accurately identify applicable pieces of the problem set. Conducting this step requires some intimate knowledge of the environment by the problem solver or designer so as to not omit critical factors that have bearing on the problem. This step is critical for the designer to move onto Step 3. Completion of this step should enable the designer to systematically create the pieces that will be used to model the design environment being investigated and aids in addressing C1.

4.1.3 Step 3: Create Modeling and Simulation (M&S) Environment

Modeling and simulation of dynamic and complex environments is never an easy task. Establishing the assumptions and obtaining agreement from all those involved is a difficult task. Building a model that mirrors the key concepts of the reality being investigated is paramount to success of a designer's results. Figure 30 below depicts the

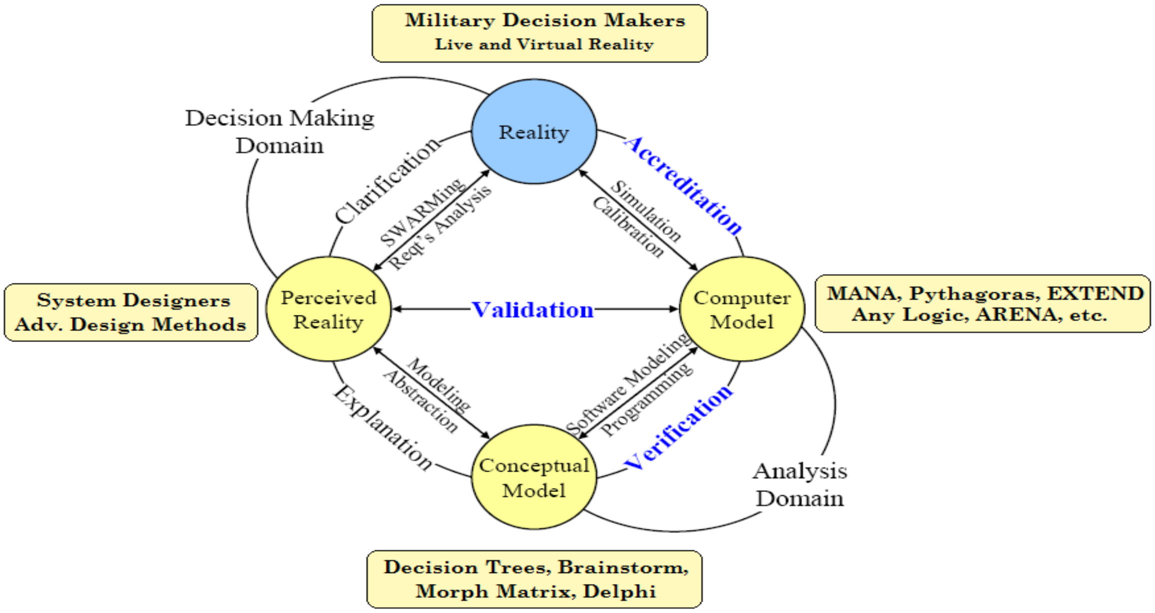


Figure 30: Modeling and Simulation Wheel (Adapted from [13])

elements and ensuing process that can help to formulate a realistic model for the design environment. Computer models have emerged as a source of creating surrogate realities for design or problem space exploration. A wide variety of these models exist in varying fidelity. Some are mathematical in form, some incorporate the capability to create environments from scratch using computer programming code such as Java or C++, while others are packaged applications created for specific industries or environments where modeling and simulation have become the norm.

Choosing the correct modeling application and involving the proper stake holders to assess the accuracy and fidelity of the model will address C1 and C2. While this is also aided by the following step, Step 4, selecting the proper application nor Step 4 aid in achieving agreement between all participants. There are many applications and techniques in gaining consensus from a group, especially when the group has competing or differing opinions. Gaining group consensus can be critical to gaining wide spread acceptability and usage by those involved in the decision making process, but is not a focus of this research.

This step will likely be an iterative one when coupled with Step 4 because agreement on the results of the model and how those results were obtained lead to the credibility of the output. Performing Step 2 adequately is also important to creating a correct model. As the reader can see meeting challenges C1 and C2 involves more than just one step. This is indicated in Figure 26 with the feedback loops between these processes that culminate with the next step in the ATLAS method before moving on in the process outlined in Figure 25.

4.1.4 Step 4: Verify, Validate, Accreditation the M&S Environment

Often times there are requirements or a desire to conduct measures of goodness a created model represents. This is most approached by conducting verification, validation, and accreditation (VV&A) activities on the model. There are a variety of definitions that encompass these three efforts, but essentially they entail the following general questions a designer can ask themselves [101]:

- Verification: Did I build the model right?
- Validation: Did I build the right model?
- Accreditation: Should this model be used, how confident are the results, and when should it be used?

This step is the culmination of the first three steps and partially addresses C5. Its intent is to make sure the problem was adequately investigated, defined, and translated into a virtual medium for exploration and or problem resolution. Verification of M&S is accomplished by observing the model under various test conditions to see if the model is acting in a prescribed or known way. In engineering this is relatively straight forward if the model is primarily based on physical aspects, equations, or quantifiable data. When qualitative or hybrid environments are involved the measure can sometimes be soft because verification usually requires only that the model satisfy the intent of creator. Criteria or specific tests are usually provided to add rigor to this process and satisfy the perceived need for those all wishing to utilize the model for an intended purpose.

Validation among many things investigates if the model possesses an adequate level of fidelity for the environment being modeled. Basic processes require simple models, but in today's environment nothing remains simple and models can begin to become complicated and almost impossible to model reality exactly. To alleviate this issue assumptions are made by modelers and this step in VV&A can be used to scrutinize these assumptions. Finally this step endeavors to measure how well the model outcomes compare to some referent data set that is accepted as representative of real or quantifiable outputs.

Accreditation is the process of deciding whether or not the model is applicable in its form to be applied to the design space. Will its use be valuable and will its use meet acceptable cost, schedule, and other criterion or time constraints being imposed on the problem environment that seeks to use the M&S applications are general criteria considered.

4.1.5 Step 5: Create / Update a Case Based Database

This step seeks to collect and arrange the outcomes of either measured real world data applicable to the problem set or the results of simulation outcomes from an applicable model. Data can be generated in extensive amounts in today's computer intensive environment. This data if not properly managed can literally bog down a system or environment such that conducting analysis or getting meaningful results is impossible. There exist many techniques and off the shelf applications that seek to help in data management endeavors. These applications are usually the result of users in the field experiencing difficulty with data management and creating software that aids in effective organization of information. Often times these solutions are able to transcend beyond their intended use and be applied in a variety of different fields that have similar needs. For this reason selection of data management applications should be scrutinized by decision makers for their applicability, capabilities, and feasibility for the desired need. This step in the ATLAS method attempts to strike a balance between several criteria. These criteria will likely vary depending on the need and are usually born from the need of a wide range of users and maintainers of the information pool.

4.1.6 Step 6: Identify & Incorporate Uncertainty in Knowledge/Information

Known known's and unknown unknowns or in simpler terms knowing what you know and knowing what you don't know is important. Not accounting for this fact or at least having awareness of this element can be detrimental to results in any M&S environment. A look at Figure 31 shows the general impact of how uncertainty in knowledge can have an impact in the risk decisions makers may have to face in vague or ill defined environments. Accurately modeling the real world and its effects is a difficult and sometimes arduous task. Assumptions are made to simplify the environment, ranges can be applied to a data point to allow some flexibility where in reality there is not.

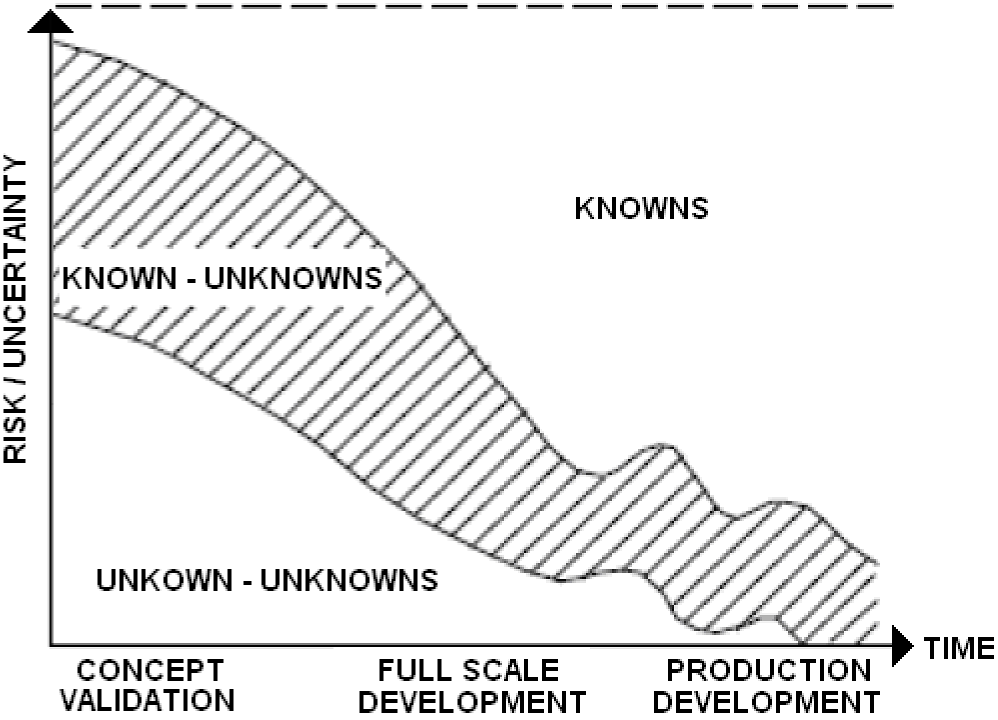


Figure 31: Risk or Uncertainty based on Knowledge Gained over Time [136]

Uncertainty incorporation and analysis can be difficult in situations where the environment is dynamically changing, time dependant, or asymmetric. To solve this dilemma many attempt to incorporate repeatability and reproducibility, and yet others try to determine fitness of information or utilize measurement metrics. This step in the process seeks to incorporate the inherent lack of knowledge in the design space input factors. This can be accomplished through a variety of quantitative or qualitative techniques. No one technique is the right solution. It is up to the designer to determine which is suitable for the problem space.

The ATLAS method uses fuzzy logic in this step to represent the decision maker's assessment of the quality or certainty of the information that represents the input variables of concern. This assessment is subjective and dependant on the DM's experiences, qualifications, and training. Using fuzzy logic adequately addresses C3 where the desire to incorporate the HITL's or the DM's cognitive assessments of information is a need identified by military leaders. The complexity and dynamics involved with real world operations require a HITL participant. Fuzzy logic allows the impact of the cognitive assessments that are unique to each and every operational environment.

4.1.7 Step 7: Create a Surrogate Model

A surrogate model or metamodel is simply a model of a model that can be used to approximate a design or problem space, generally at a computationally cheaper price. Often times, running detailed and or complex models can be time intensive or cost prohibitive to designers. As a result, tasks such as design space exploration, sensitivity analysis, and *what-if* exploration become impossible since they require sometimes thousands or even millions of simulation runs. To alleviate this quandary, surrogate models were developed. There are varying levels of fidelity for surrogate models to include linear approximations, quadratic equations, polynomial response surface equations (RSE), artificial neural networks, etc. These models usually work in a black box environment were the user does not really care how the results are achieved, but requires a medium that takes inputs and produces valid and suitable outputs. The accuracy of these models depends on how many design points or cases exist in the design space and can be created with a variety of experiments or DOE to help formulate the surrogate model. It is important to note that these models are generally only capable of modeling the design space outlined in a DOE and usually have erroneous results when applied outside this design space. There exists no strict rule for determining the best metamodel type to be used and application of surrogate types is strictly dependant on the design space. Linear, quadratic, and RSE generally perform well in smooth, linearly approximated, and continuous design spaces. ANN's perform well in complex, discontinuous, and highly nonlinear design spaces. Whatever the designer's choice, the ATLAS method seeks to incorporate a metamodel to allow design space exploration and *what-if* analysis at the user level.

This *what-if* analysis can also help conduct optimization analysis on the design space to find the best operational architecture and logistics package to support the operational environment being modeled. This analysis helps to meet C7 where optimization no longer fulfills the desires for robustness. This especially so when the operational environment is dynamic or the requirements are in flux and a solution must be adaptable to the evolving environment/design space.

4.1.8 Step 8: Visualize and Explore the Design Space

Visualization and exploration of the design space can be performed in a variety of ways in today's M&S world. The goal should be to provide these capabilities with reference to the audience viewing the environment. As stated earlier data is all around us in many fashions and managing this data is an important task, but also being able to visualize data and manipulate it in a dynamic fashion is a necessity for the fast moving pace designers and many decision makers face today. Tools such as JMP from SAS seek to provide links to statistical and output data that can be dynamically viewed through a variety of graphics. Other options exist with custom codes created in a variety of programming languages or one can use off the shelf software applications like MatLab, Java, and C++ that allow a user to perform mathematical calculations, create GUI's, and create models. The options are endless in this arena and can be as unique, extravagant, or plain as the designer desires. The ATLAS method contends that a picture is worth a thousand words. Incorporating visualization and design space exploration tools are critical to bringing information to the user quickly and in a more efficient manner.

Accomplishing this step adequately helps to address C4 and C9 and can provide information to a variety of decision makers at differing levels in support of decision making activities. The M&S environment along with a variety of data representation and visualization techniques in the following step provides a venue for scoping the sea of data prevalent in these dynamic and unpredictable environments so that a COA can be developed with foundational support data. If the designer has adequately performed the steps outlined in Figure 25 and gained agreement from stakeholders, the cost and feasibility of usage by decision makers should be a much easier task,

4.1.9 Step 9: Determine COA for Architecture Selection or Iterate Back to Step 5 or Step 1 if Necessary

Now that the decision maker has visualized and explored the design space, the goal is that useful information has been garnered by the user or decision maker in such a way that a decision support system has been created that can provide justification for a particular course of action. With the copious amounts of data available, no one human can adequately process all of it. Computers can help with complex and intensive mathematical calculations, as well as allow virtual representation of the design space. This final step of the ATLAS method now calls upon the HITL or decision maker to use the post processed information acquired by the ATLAS method to determine a potential solution. The ultimate question should be is there enough information for the decision maker to begin determining and coming to a solution in terms of logistical needs and an adequate support architecture. This step addresses C5 in the determination of sufficiency of information to finding a relevant COA to pursue. If a suitable solution cannot be reached the designer has the option of reverting back to previous steps to better define or model the design space. Answering C5 arguably is a subjective assessment and will be significantly aided by the efforts in Step 4.

4.2 Summary

The proposed methodology aims to bring together several areas of research that have garnered renewed interest or created for the purpose of solving highly nonlinear and dynamic design problems. Case Based Reasoning, fuzzy logic, and artificial neural networks will be incorporated together under the ATLAS method. CBR has shown significant promise in finding relevant and reasonable solutions/answers when large amounts of data and complex relationships between input factors are involved. Fuzzy logic has reemerged, thanks largely to advances in computing, and is being applied to a variety of problems where human in the loop, linguistics, and the need for a balance between high fidelity and imprecision are beneficial. Finally, ANN have shown the unique ability to decompose complex and nonlinear environments to aid in producing surrogate model representations of these real world environments.

The combination of these three core concepts should be beneficial in the previously described military logistics arena where data is often times scarce, ambiguous, or just not readily available. Overcoming this gap and formulating a way to solve the military's logistics requirements by developing a method that performs better than antiquated linear methods should help to advance and further areas were predictive analysis is a key aspect to success. This should allow those who need to have more assurance than a "gut" feel about the future outcome and performance of real world systems to have justifiable and a basis for specific course of action selections.

5 PROOF OF CONCEPT

> *"Things should be as simple as possible, but not simpler"*
> -Albert Einstein

5.1 Establishing the Scenario

The scenario applied to the ATLAS method is based upon a scenario generated by the USMC and was subsequently adapted by CPT Matthew Bain (USMC) for his Masters Thesis while attending the Naval Postgraduate School. This scenario was established by the Marine Corps War-fighting Lab (MCWL) for a Distributed Operations Seminar War-game held in Nov 2004 and employs a platoon sized element, depicted in Figure 32. The mission for the platoon is to conduct interdiction and surveillance operations across a six kilometer border in a region of rugged terrain for seven days.

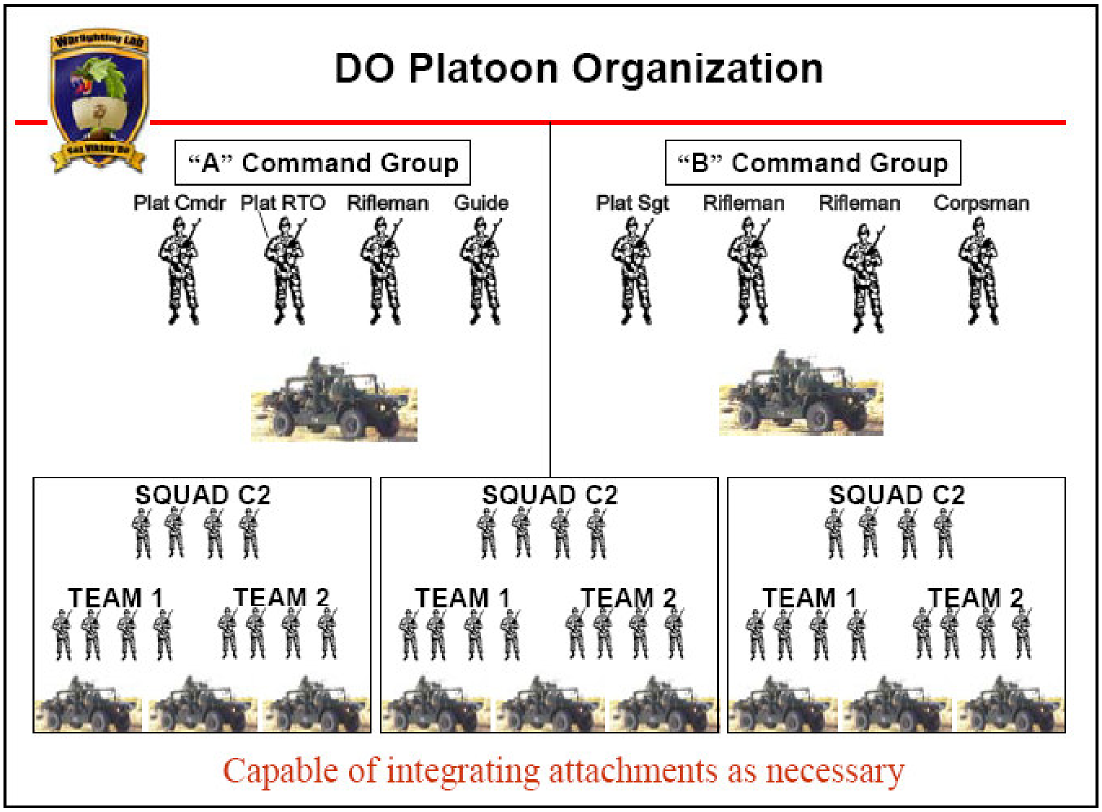


Figure 32: DO Platoon Organization [24]

The operational scenario is one in which an unknown sized enemy force uses part of a border area for staging, link up points, and entrance and exit points for routes along the six kilometer border. Intelligence has provided the likely enemy COA in Figure 33.

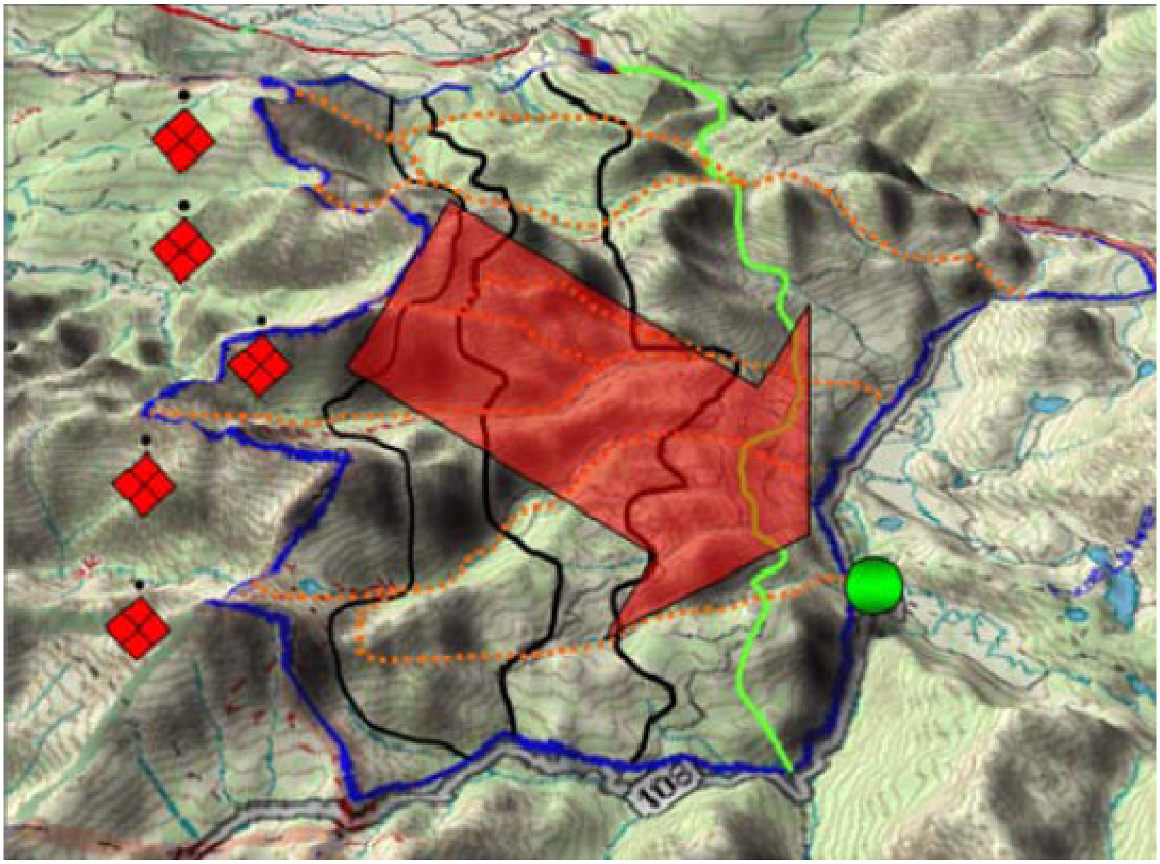


Figure 33: Intelligence for Most Likely Enemy COA [114]

In support of the platoon for this mission are aviation delivery assets that can deliver supplies vicinity of the platoon's distributed locations. The platoon will be split up into its three squads and two command elements to provide effective coverage of the area. The combat support structure (Seabase) is approximately 50 km away. Available to the platoon are a variety of communication assets for internal and external communication with 100% reliability. Also available to the platoon are indirect fire assets in the form of an artillery battery for either defensive or engagement operations. The geographical depiction is in Figure 34.

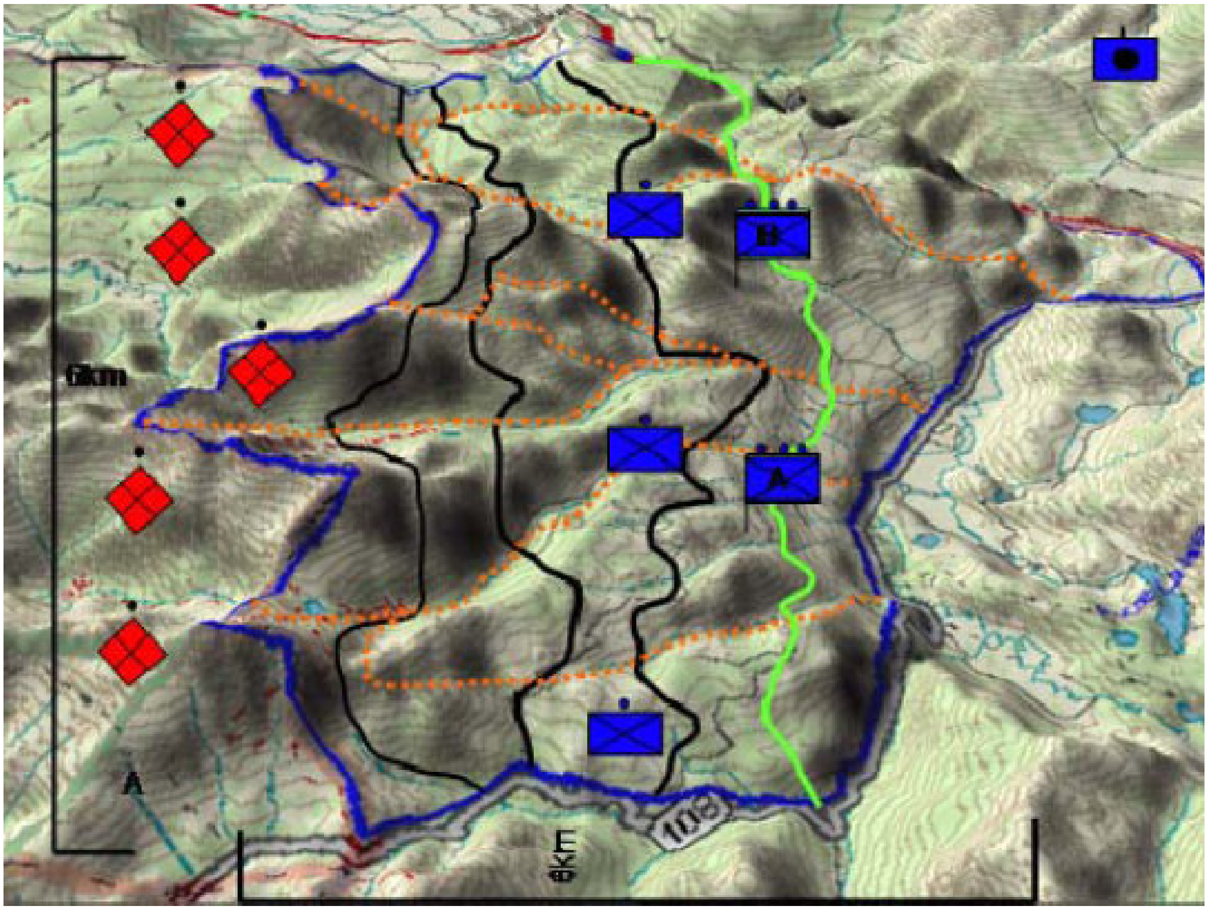


Figure 34: Operational Area Geography [114]

Analyzing combat operations and building subsequent models that represent this space involves accounting for intangibles along with any of the physical aspects that can be mathematically represented. Intangibles can primarily be defined as those attributes that a soldier or enemy combatant possesses while operating in the operational area. These traits consist of, but are not limited to morale, courage, ingenuity, determination, and luck. The development of this scenario involved many people. Some of the people on the list of contributors are experts from the military with years or operational experience, researchers from the Navy, Marine Corp, and Army who have worked extensively in the analysis and development of tools and methodologies aimed at giving a clearer picture when it comes to logistics, along with others that have had some experience in the application of software and modeling tools. For that reason the author determined, through his own dissection and military experience of 13 years that the scenario and base model developed via CPT Bain's research was a good starting point.

The basic scenario above contains the objective, mission, and intent for the DO platoon's logistic support requirements being analyzed. It has been established that supporting this platoon will be critical to its success. Without proper supporting elements and a well thought out architecture the platoon will likely be incapable of successfully performing its primary or any alternate missions. Organic to this platoon are weapons, vehicles, radios, medical supplies that all require consumables like ammunition, batteries, fuel, etc to keep operating. Other consumables also consist of basic sustenance of food and water for each soldier. All of these commodities are consumed at varying rates depending on the situation. In the past the military determined supply quantities based on a per man metric that was determined by aggregating historical supply data. As has been discussed, the military is moving away from a large supply depot ashore to reduce their logistics footprint and promote flexibility and robustness of their forces capability to react to varying operational scenarios. The solution to these requirements, the military feels, is Seabasing. As such determining the architecture of a Seabase and its components must be determined. In order to do this, a fundamental understanding of the forces to be supported like a DO platoon should be investigated.

5.2 Traditional Logistics Determination

As has been discussed, the military is keenly aware of their logistics woes and is currently trying to augment/increase their ability to perform logistics activities in a seamless and transparent manner. Incorporating new and relevant operational data is the best way to see emerging trends and forecast the logistical needs for the changing force. The DO Platoon modeled in MANA can be compared to that of a specialized Army Infantry Ranger platoon. They are almost exactly the same in number of soldiers, capabilities, and force makeup in terms of weapons, communications capabilities, and vehicle assets. For that reason this Army unit will be used as a comparable force for application to the Army's logistics planning tool discussed next.

5.2.1 Operations Logistics Planner (OPLOGPLN)

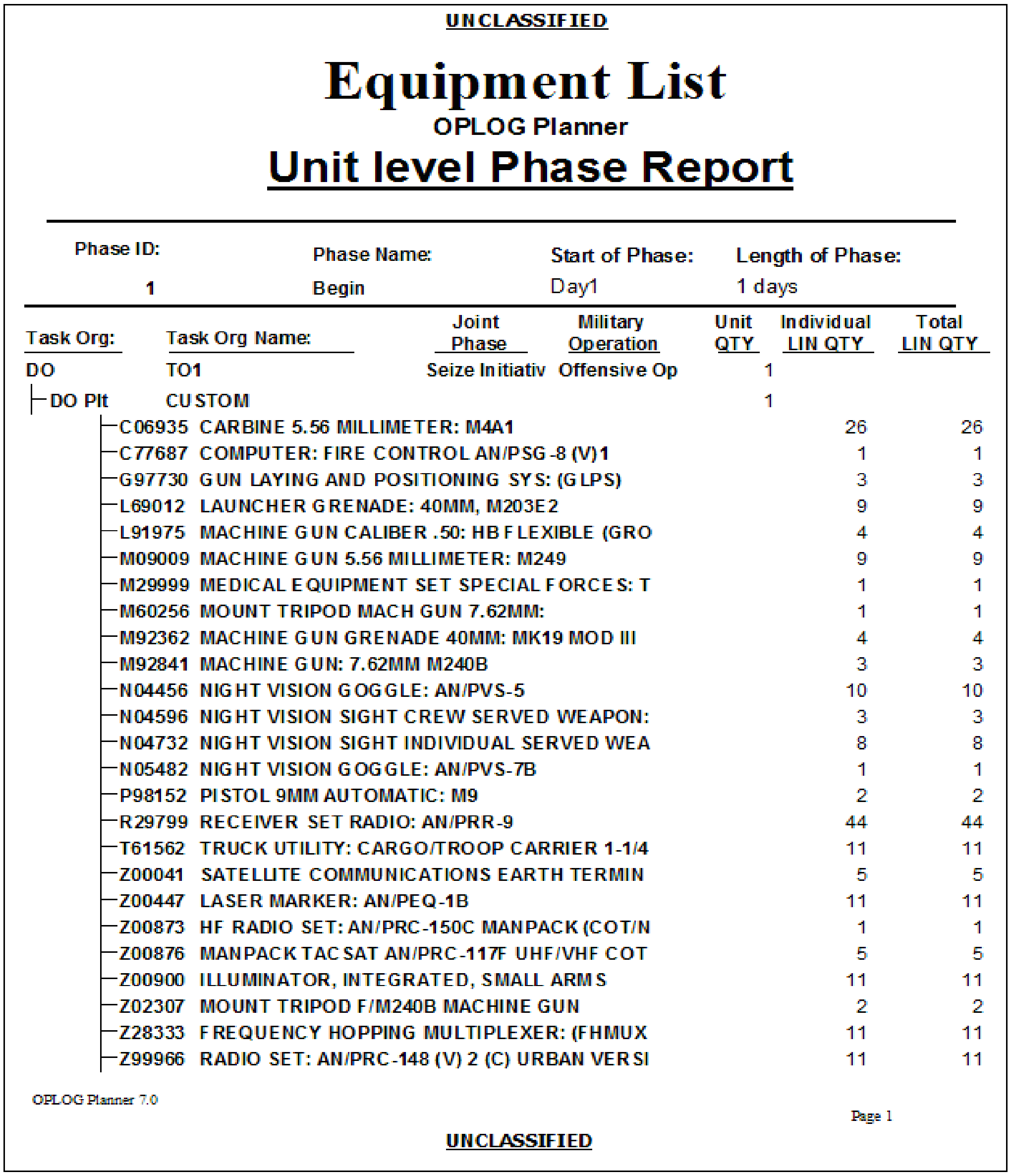
The Army has developed a tool called Operational Logistics Planner (OPLOGPLN) that serves as a logistics planning tool for Army units and organizations currently in existence today, see Section 2.14.1. OPLOGPLN is a Microsoft Access Database GUI available from the Combined Arms Service Command (CASCOM) located at Ft. Lee, Virginia. The user is referred to [22] for more information on the background and how to use this program. The program consists of six processes that the user performs to determine the logistics need for a specified unit. These steps are:

Step 1: Select Units
Step 2: Create Task Organization
Step 3: Determine Operational Phases
Step 4: Set consumption Parameters
Step 5: Select Reports to Create
Step 6: View Available Reports

Table 5: Equipment List for Command Group A/B Elements & Squads [8]

Equipment	Description	Quantity
Adv Med Kit	Advanced Medical Kit	1
ETCS	Expeditionary Tactical Communication System	5
FHMUX	Frequency Hopping Multiplexer	11
GLTD	Ground Laser Track Designator	3
ITV	Internally Transportable Vehicle or HMMWV	11
IZLID	Infrared Zoom Laser Illuminator/Designator	11
M4A1	5.56 mm Rifle	26
M-203	5.56 mm Rifle with Grenade Launcher	9
M-240G	7.62 Machine Gun w/ Tripod	3
M-249	5.56 mm Machine Gun	9
M-2HB	0.50 Caliber Machine Gun	4
M-9	9 mm Pistol	2
Med Kit	Standard First Aid Kit	5
MK-19	Automatic 40 mm Grenade Launcher	4
PAS 13-H	Optic for 0.50 Caliber Machine Gun or MK-19	8
PEQ-2	Pointing Laser	11
PRC 117	Squad ,Platoon, HHQ Fire Control Digital Communications	5
PRC 148	Platoon Command and Control and Close Air Support Radio	11
PRC 150	Platoon to HHQ Communications	1
PRR	Personal Roll Radio	44
PVS-17	Night Sight for M-240G	3
PVS-5	Night Vision Personal	11
Soldiers	Self Explanatory	44

The first step allows the user to either select premade generic Army units that currently exist under the military structure or the user can create a custom unit from an extensive equipment list to create a Modified Table of Organization and Equipment (MTOE) that will define the custom unit. It is important to note that when a custom unit is created, clearly defining the assets and force makeup are the foundation for subsequent steps where resource allocation is aligned with the custom unit's MTOE. Referring to Figure 32, the makeup of the Ranger platoon in OPLOGPLN consists of the items in Table 5 and as indicated by Bain in his work. Initiating OPLOGPLN's ability to create custom units, the author created a 44 man platoon with the equipment listed in Table 5. Because the Army and Marines are primarily ground forces, it was not extremely difficult to create the units to match according to the equipment list. What was difficult was filtering through the extensive equipment database associated with OPLOGPLN as well as dealing with duplicate entries of equipment types under varying similar nomenclature or descriptions. A partial report of the makeup of the platoon created in OPLOGPLN is displayed in Figure 35. The full report is in Appendix F.



UNCLASSIFIED

Equipment List

OPLOG Planner

Unit level Phase Report

Phase ID:	Phase Name:	Start of Phase:	Length of Phase:
1	Begin	Day1	1 days

Task Org:	Task Org Name:	Joint Phase	Military Operation	Unit QTY	Individual LIN QTY	Total LIN QTY
DO	TO1	Seize Initiativ	Offensive Op	1		
DO Plt	CUSTOM			1		
C06935	CARBINE 5.56 MILLIMETER: M4A1				26	26
C77687	COMPUTER: FIRE CONTROL AN/PSG-8 (V)1				1	1
G97730	GUN LAYING AND POSITIONING SYS: (GLPS)				3	3
L69012	LAUNCHER GRENADE: 40MM, M203E2				9	9
L91975	MACHINE GUN CALIBER .50: HB FLEXIBLE (GRO				4	4
M09009	MACHINE GUN 5.56 MILLIMETER: M249				9	9
M29999	MEDICAL EQUIPMENT SET SPECIAL FORCES: T				1	1
M60256	MOUNT TRIPOD MACH GUN 7.62MM:				1	1
M92362	MACHINE GUN GRENADE 40MM: MK19 MOD III				4	4
M92841	MACHINE GUN: 7.62MM M240B				3	3
N04456	NIGHT VISION GOGGLE: AN/PVS-5				10	10
N04596	NIGHT VISION SIGHT CREW SERVED WEAPON:				3	3
N04732	NIGHT VISION SIGHT INDIVIDUAL SERVED WEA				8	8
N05482	NIGHT VISION GOGGLE: AN/PVS-7B				1	1
P98152	PISTOL 9MM AUTOMATIC: M9				2	2
R29799	RECEIVER SET RADIO: AN/PRR-9				44	44
T61562	TRUCK UTILITY: CARGO/TROOP CARRIER 1-1/4				11	11
Z00041	SATELLITE COMMUNICATIONS EARTH TERMIN				5	5
Z00447	LASER MARKER: AN/PEQ-1B				11	11
Z00873	HF RADIO SET: AN/PRC-150C MANPACK (COT/N				1	1
Z00876	MANPACK TAC SAT AN/PRC-117F UHF/VHF COT				5	5
Z00900	ILLUMINATOR, INTEGRATED, SMALL ARMS				11	11
Z02307	MOUNT TRIPOD F/M240B MACHINE GUN				2	2
Z28333	FREQUENCY HOPPING MULTIPLEXER: (FHMUX				11	11
Z99966	RADIO SET: AN/PRC-148 (V) 2 (C) URBAN VERSI				11	11

OPLOG Planner 7.0

Page 1

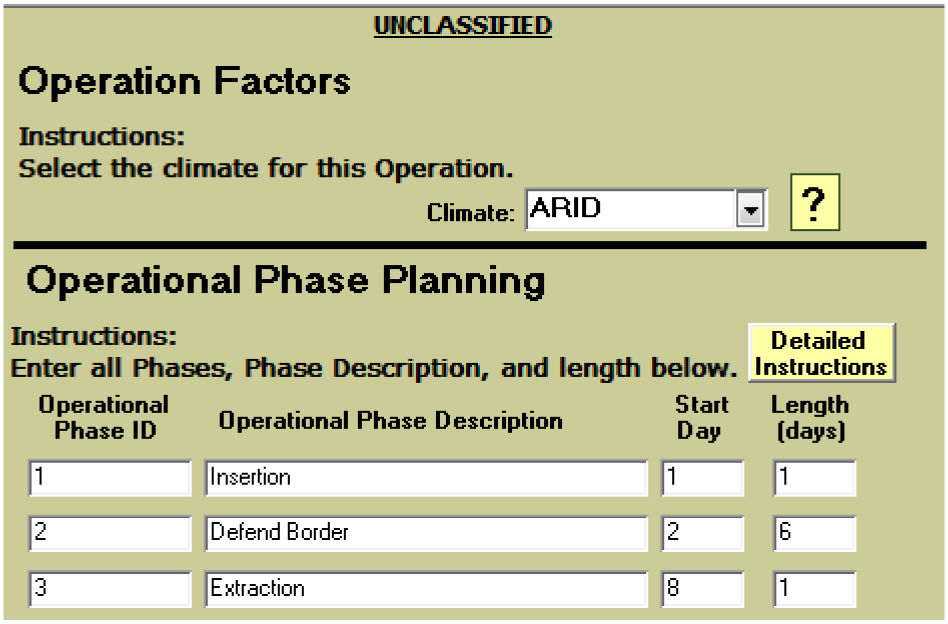
UNCLASSIFIED

Figure 35: OPLOGPLN Unit Level Equipment List

The only line item the author was unable to add because it was not found in the OPLOGPLN database is the soldier level medical kit. Since the weight of this item is not critical for this particular analysis this was deemed negligible. Continuing through the process, executing OPLOGPLN next requires the user to setup the Operation Factors and Operational Phase Planning. The user has four options when selecting the climate conditions the force will be operating in. They are listed below with definitions for each climate condition. The condition selected for this analysis was Arid.

- Arctic: a climate of generally treeless zones of tundra and of the regions of permafrost in the Northern Hemisphere.
- Arid: a dry climate with an annual precipitation usually less than 10 inches.
- Temperate: a climate which is never extremely hot or extremely cold
- Tropic: a climate with continually high temperatures with considerable precipitation, at least during part of the year.

The Operational Phase Planning is dictated by the user by separating the operation into as many phases as the user feels is necessary. The author specified three phases for this scenario: insertion - day one, interdiction activities along the six kilometer border - seven days, and then finally extraction from the operational area - day eight, see Figure 36.



UNCLASSIFIED

Operation Factors

Instructions:
Select the climate for this Operation.

Climate: ARID

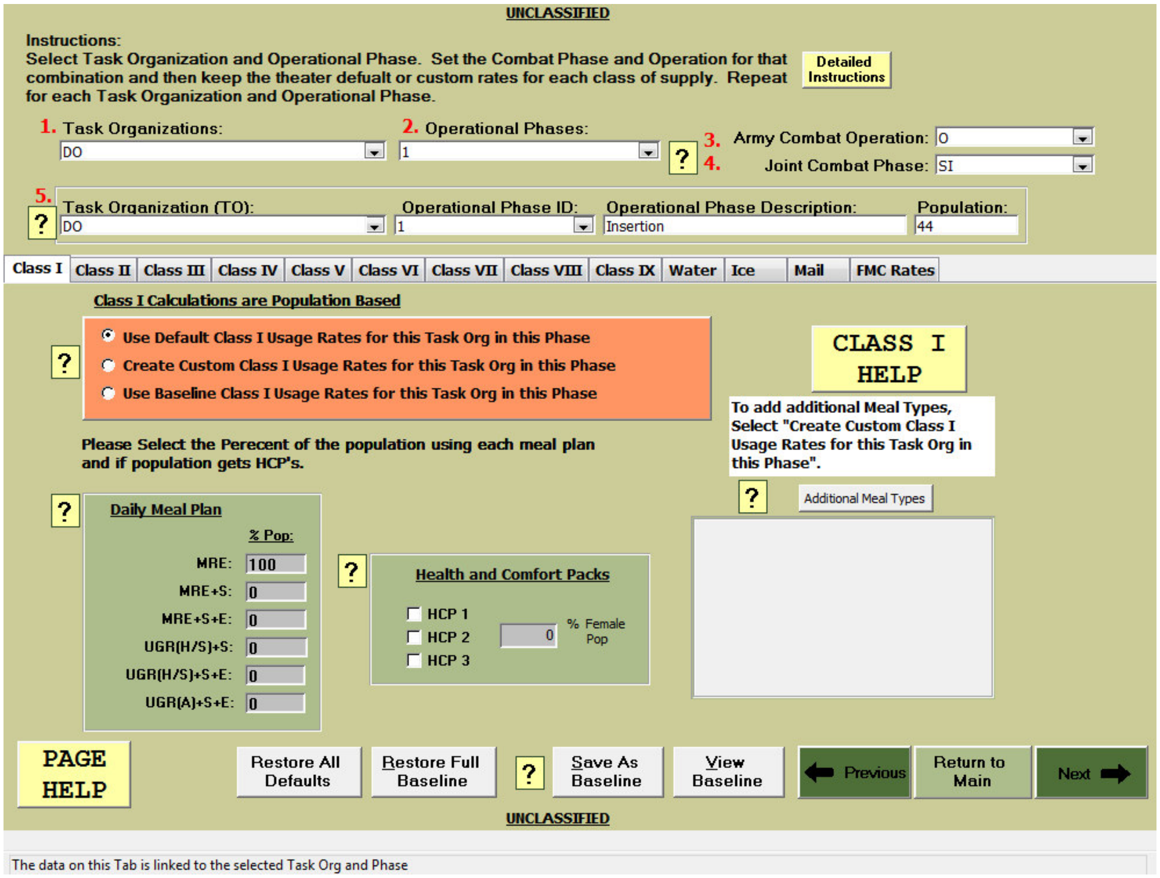
Operational Phase Planning

Instructions:
Enter all Phases, Phase Description, and length below.

Operational Phase ID	Operational Phase Description	Start Day	Length (days)
1	Insertion	1	1
2	Defend Border	2	6
3	Extraction	8	1

Figure 36: Scenario Operational Characterization

Next the author set the consumption parameters for the DO platoon for each of the three phases. The screen the user sees is shown in Figure 37. These three phases are further separated into Army Combat Operation and Joint Combat Phase settings, defined in Appendix B4. In the consumption parameter section, the user can specify a variety of consumption and usage rates for the unit via the nine classes of military supply. Several of the classes, specifically Class V, VII, and IX, are determined based on historical data and or are classified so they are calculated in a black box environment. The characterization of the scenario was designated as Offensive Operations for the Army Combat Operations Phases and Dominate for the Joint Combat Phase for all phases of the scenario. Meals Ready to Eat (MRE) were the sole source of subsistence (3 MREs/day)



UNCLASSIFIED

Instructions:
Select Task Organization and Operational Phase. Set the Combat Phase and Operation for that combination and then keep the theater defualt or custom rates for each class of supply. Repeat for each Task Organization and Operational Phase.

1. Task Organizations: DO
2. Operational Phases: 1
3. Army Combat Operation: O
4. Joint Combat Phase: SI
5. Task Organization (TO): DO
Operational Phase ID: 1
Operational Phase Description: Insertion
Population: 44

Class I Calculations are Population Based

● Use Default Class I Usage Rates for this Task Org in this Phase
○ Create Custom Class I Usage Rates for this Task Org in this Phase
○ Use Baseline Class I Usage Rates for this Task Org in this Phase

Please Select the Perecent of the population using each meal plan and if population gets HCP's.

To add additional Meal Types, Select "Create Custom Class I Usage Rates for this Task Org in this Phase".

Daily Meal Plan

	% Pop:
MRE:	100
MRE+S:	0
MRE+S+E:	0
UGR(H/S)+S:	0
UGR(H/S)+S+E:	0
UGR(A)+S+E:	0

Health and Comfort Packs

☐ HCP 1
☐ HCP 2
☐ HCP 3

% Female Pop: 0

UNCLASSIFIED

The data on this Tab is linked to the selected Task Org and Phase

Figure 37: OPLOGPLN Consumption Parameters

and the water planning factor equated to 4.84 gallons/day/man. The final screen, not shown, allows the user to select the specific reports the user would like OPLOGPLN to generate. These reports cover each of nine classes of supply in detail and also include mail, water, and ice forecasts. The consolidated logistics forecast from each report is shown in Table 6. The full reports generated by OPLOGPLN can be seen in Appendix F

Table 6: Consolidated OPLOGPLN Logistical Data Outputs.

Supply	Weight (lbs)			
	Total	Phase 1	Phase 2	Phase 3
Class I	1932.48	241.56	1449.36	241.56
Class II	728.2	158.4	488.4	81.4
Class III	145.97	18.24	109.47	18.24
Class III (Fuel)	2540.95	317.61	1905.71	317.61
Class IV	823.67	102.96	617.7599	102.96
Class V	809.42	101.17	607.07	101.17
Class VI	197.12	24.64	147.84	24.64
Class VII	52.8	6.6	39.6	6.6
Water	14205.82	1775.71	10654.32	1775.71

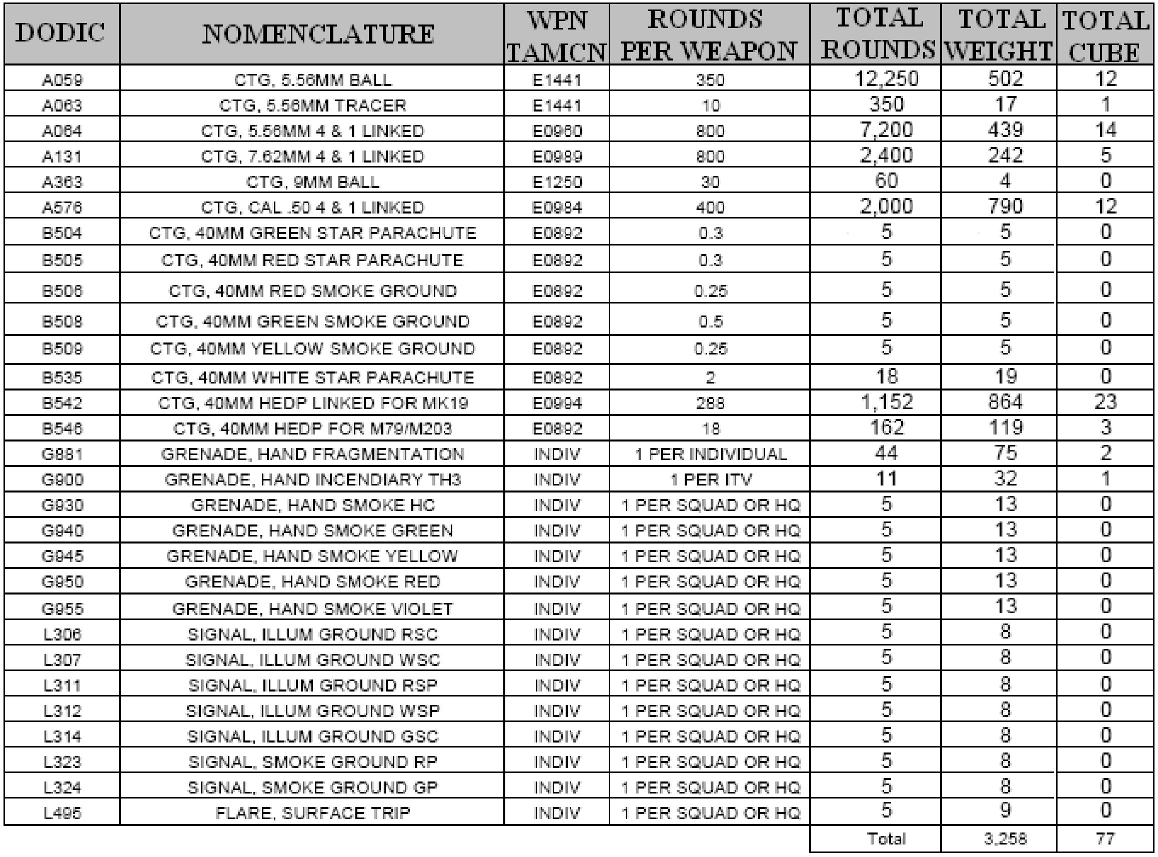
Planning factors for the above classes from the CASCOM website are shown in Table 7.

Table 7: Current Planning Factors for Army Logistics Planning via CASCOM

Supply Class	Pound/day/man	Supply Class	Pound/day/man
Class I	5.49	Class V	Classified & only at Theatre Level
Class II	NE Asia: 1.85 SW Asia: 1.52	Class VI	NE Asia: 1.10 SW Asia: 1.30
Class III	Varies depending on Organization	Class VII	Varies depending on Organization
Class IV	4.37	Water	Temperate: 2.90 Arid: 4.50

The total Class I weight is a simple calculation based on 3 meals/day/man and accounts for 1056 MRE's for the 44 man platoon for eight days. Class II is also calculated by taking the number of days and multiplying by the planning factor to get the total for the platoon over the eight days. This is primarily done across the board, except for Class V, III, and VII, which either vary on the organizations MTOE or data on recent usage rates, this is especially so in terms of ammunition and ordnance where 2005 data is used to calculate Class V forecasts. Discussions with CASCOM also revealed that this tool primarily looks at major combat operations and aggregates information and data to determine logistical forecasts for military planners. CASCOM is working on creating tools for the lower level logistician because the realization is that planning, especially for Class V, must be a bottom up process and not a top down process. The OPLOGPLN output of Class V, is critical and just as if not more important than Class I and water are to a soldier's survival in a combat environment. The resulting forecast of Class V by OPLOGPLN is that it predicts 809 pounds of total ammunition for the entire 8 day scenario is required; a significant shortage based on work conducted by Bain. A look at Table 8 and the calculations for a typical combat load for a DO platoon type unit shows that for three days of supply, the ammunition weight alone comes to 3258 pounds.

Table 8: Basic Combat Load for a DO Platoon [8]



DODIC	NOMENCLATURE	WPN TAMCN	ROUNDS PER WEAPON	TOTAL ROUNDS	TOTAL WEIGHT	TOTAL CUBE
A059	CTG, 5.56MM BALL	E1441	350	12,250	502	12
A063	CTG, 5.56MM TRACER	E1441	10	350	17	1
A064	CTG, 5.56MM 4 & 1 LINKED	E0960	800	7,200	439	14
A131	CTG, 7.62MM 4 & 1 LINKED	E0989	800	2,400	242	5
A363	CTG, 9MM BALL	E1250	30	60	4	0
A576	CTG, CAL .50 4 & 1 LINKED	E0984	400	2,000	790	12
B504	CTG, 40MM GREEN STAR PARACHUTE	E0892	0.3	5	5	0
B505	CTG, 40MM RED STAR PARACHUTE	E0892	0.3	5	5	0
B506	CTG, 40MM RED SMOKE GROUND	E0892	0.25	5	5	0
B508	CTG, 40MM GREEN SMOKE GROUND	E0892	0.5	5	5	0
B509	CTG, 40MM YELLOW SMOKE GROUND	E0892	0.25	5	5	0
B535	CTG, 40MM WHITE STAR PARACHUTE	E0892	2	18	19	0
B542	CTG, 40MM HEDP LINKED FOR MK19	E0994	288	1,152	864	23
B546	CTG, 40MM HEDP FOR M79/M203	E0892	18	162	119	3
G881	GRENADE, HAND FRAGMENTATION	INDIV	1 PER INDIVIDUAL	44	75	2
G900	GRENADE, HAND INCENDIARY TH3	INDIV	1 PER ITV	11	32	1
G930	GRENADE, HAND SMOKE HC	INDIV	1 PER SQUAD OR HQ	5	13	0
G940	GRENADE, HAND SMOKE GREEN	INDIV	1 PER SQUAD OR HQ	5	13	0
G945	GRENADE, HAND SMOKE YELLOW	INDIV	1 PER SQUAD OR HQ	5	13	0
G950	GRENADE, HAND SMOKE RED	INDIV	1 PER SQUAD OR HQ	5	13	0
G955	GRENADE, HAND SMOKE VIOLET	INDIV	1 PER SQUAD OR HQ	5	13	0
L306	SIGNAL, ILLUM GROUND RSC	INDIV	1 PER SQUAD OR HQ	5	8	0
L307	SIGNAL, ILLUM GROUND WSC	INDIV	1 PER SQUAD OR HQ	5	8	0
L311	SIGNAL, ILLUM GROUND RSP	INDIV	1 PER SQUAD OR HQ	5	8	0
L312	SIGNAL, ILLUM GROUND WSP	INDIV	1 PER SQUAD OR HQ	5	8	0
L314	SIGNAL, ILLUM GROUND GSC	INDIV	1 PER SQUAD OR HQ	5	8	0
L323	SIGNAL, SMOKE GROUND RP	INDIV	1 PER SQUAD OR HQ	5	8	0
L324	SIGNAL, SMOKE GROUND GP	INDIV	1 PER SQUAD OR HQ	5	8	0
L495	FLARE, SURFACE TRIP	INDIV	1 PER SQUAD OR HQ	5	9	0
				Total	3,258	77

For the DO platoon, sustained operations over the entire mission are expected. While supply usage will really depend on how often enemy contact occurs, 809 pounds does not even equate to a single days expected usage based on combat standard planning factors, let alone a whole week of engagements.

Although faced with this shortcoming, a look at the OPLOGPLN Class V report (Appendix F 6) when compared to Table 8 does show the same line items and ammunition type and class. This gave the author some verification that the Army Ranger platoon created in OPLOGPLN was setup correctly within the tool. The OPLOGPLN tool in the end provides reports on averaged logistics for the unit specified, but is incomplete. Even if it were capable of accurately forecasting logistical support requirements in terms of consumables for the designated small unit, the tool does not provide any information on suitable resupply package sizes, when they will need to be delivered, or allow the user to incorporate intelligence or data gathered at the lower level where small unit operation will be taking place and where determination of an adequate logistics support architecture will be conducted.

5.3 The ATLAS Method

Utilizing the same scenario that has been developed for analysis in this thesis consisting of a DO Platoon deployed for approximately seven days. A user can enlist the aid of the ATLAS tool GUI, to be discussed in Section 5.3.8.2, this tool facilitates the information and activities resulting from Steps 1 through 7 of the ATLAS method. This tool provides an easy to use interface for the DM, planner, and or logistician at the operational level that supports the DO platoon. At this level, the logistician for a MEU and the likely parent unit for a DO Platoon, would be responsible for supporting the DO platoon. This person will have intimate knowledge of what resources are and aren't available to the MEU to perform logistic support activities.

The MEU has at its disposal a complement of aerial, ground, and other vehicle assets that enable it to provide logistical support in a number of ways. Like any other manager of systems, it is the MEU's logistician's responsibility to manage these assets efficiently. His/her actions are highly correlated to the individual units in the MEU succeeding or failing in their assigned missions. An MEU is typically made up of the assets listed in Table 9. The critical assets that will be utilized for supporting the DO Platoon are highlighted in yellow. The Aviation Combat Element (ACE) and Logistics Combat Element (LCE) consists of support units within the MEU and play a critical role in supporting units like the DO platoon. The ACE is a Marine Composite Squadron

Table 9: MEU Table of Organization &Equipment (TO&E) [75]

Quantity	Nomenclature	Use
30	Medium Tactical Vehicle Replacement trucks	multiple
63	High Mobility Multipurpose Wheeled Vehicle	multiple
7	500 gallon water containers	multiple
2	TX51-19M Rough Terrain Forklift	logistics
4	Tractor, Rubber Tire, Articulated Steering	logistics
2	Reverse Osmosis Water Purification Unit	logistics
4	Mk48 Logistics Vehicle System	logistics
1	Medium Tactical Vehicle Replacement dump truck	logistics
1	LMT 3000 water purification unit	logistics
3	D7 bulldozer	logistics
8	M252 81mm mortar	ground
4	M1A1 main battle tank	ground
7 to 16	Light Armored Vehicle	ground
8	FGM-148 Javelin anti-tank missile	ground
8	BGM-71 Tube Launched, Optically Tracked, Wire Guided (TOW) Missile	ground
15	Amphibious Assault Vehicle	ground
6	155mm howitzer: M198 or M777	ground
3	UH-1N Twin Huey utility helicopter	aviation
2	KC-130 Hercules re-fueler/transport aircraft	aviation
6	CH-53E Super Stallion heavy lift assault helicopter	aviation
12	CH-46E Sea Knight medium lift assault helicopter	aviation
6	AV-8B Harrier jet	aviation
4	AH-1W Super Cobra attack helicopters	aviation

(Reinforced) composed of a Medium/Heavy Helicopter Squadron that is augmented with three other types of helicopters, one detachment of amphibious flight-deck-capable jets, and a Marine Air Control Group Detachment with air traffic control, direct air support, and anti-aircraft assets [75]. The LCE contains all the logistics specialists and equipment necessary for the unit to support itself for 15 days in an austere expeditionary environment [75]. It includes service support, medical, dental, maintenance, transportation, explosive ordinance disposal, military police, utilities production and distribution, bulk fuels, and other technical experts [75]. Of note to the equipment list in Table 9 is that the MV-22 Osprey is in the process of replacing the CH46E Super Night and the AH-1Z is scheduled to replace the AH1-W (2009) in the MEU's TO&E. The DO platoon scenario will now explored with these changes and with the ATLAS method.

5.3.1 Step 1: Define the Problem

Step 1 requires a problem definition, in this case the logistician along with direction from senior leader and peers will develop a landscape of the operational environment based on varying levels on intelligence available. The base scenario is being utilized with the DO platoon being deployed upon a six kilometer border with the goal of preventing enemy combatant access across the border. The logistician can now gather information he has available about the situation, some of which will eventually be used to codify the operational environment. This information transformed into operational variables can be categorized into three classes and two types. The first type of variable consists of factors that are controllable by the logistician. This type of information has two classes, information about the platoon and its capabilities and information about the logistics support element and aviation assets. This class and two types of data will be determined by internal status reports, known operational capabilities, or from standard operating procedures within the MEU and its units. The second type of information encompasses uncontrollable factors and falls into the third class of data that describes the enemy force, its perceived capabilities, actions, and or intent. These factors will likely be determined by intelligence and will contain some level of uncertainty and variability, especially as the situation develops.

5.3.2 Step 2: Identify Systems Architecture and Major Components

The DO platoon envisioned has a basic architecture as outlined in Appendix B 1. The two tables in this section list all of the equipment and assets available to the platoon as well as the structural break down for command and control purposes. The biggest advantage the platoon brings to the table is its ability to communicate effectively, sense its surrounding through enhanced sensors (night vision equipment, satellite imagery, and other forms of intelligence), and equipment capable of identifying and engaging an enemy at great distances. With these technology-enabled enhancements arise requirements to support the platoon in a variety of ways that historically were only needed for large military units. Unfortunately, aggregation can no longer be utilized to determine the logistical requirements. While the consumables needed to support the DO platoon are rather generic and generally readily available, large scale operations made up of many distributed forces will surely reduce the availability of these resources. This increase in support requirements, if not forecasted properly, would likely result in these consumables being in short supply; hence the purpose of this research and the development of the ATLAS methodology. Ultimately, information about the operational environment and ways for a logistician to assess the need of the engaged forces is needed.

Filling this gap should help the MEU realize efficiencies, aid in proper support architecture selection, and facilitate the robustness of the MEU support elements.

The MEU, as alluded to in Table 9, has a wide variety of vehicles as part of the table of organization and equipment (TOE). Table 10 shows the relevant vehicles for this analysis with the CH-46E replaced by the MV-22 Osprey and the AH1-W replaced by the AH1-Z Viper. Within the MEU, the logistics workhorse, especially in a seabased environment, will be the MV-22 and the CH53E. These two aircraft are often times supplemented with two AH-1 attack helicopters for defensive purposes. The addition of the AH-1 is certainly not a requirement as the MV-22 is equipped with a 7.62 mm machine gun at the rear of the aircraft and is being looked at for retrofit of a triple barrel 0.50 caliber Gatling gun for the front of the aircraft. The CH53E also can be adequately armed for defensive purposes with at least two side-mounted and one rear-mounted 0.50 caliber machine guns [21], [92]. The capabilities and basic performance characteristics of the MV-22, CH53E, and AH-1Z aircraft are shown in Table 11.

Table 10: DO Platoon Specific Equipment for Logistics Support

Quantity	Nomenclature	Use
63	High Mobility Multipurpose Wheeled Vehicle	multiple
3	UH-1N Twin Huey utility helicopter	aviation
6	CH-53E Super Stallion heavy lift assault helicopter	aviation
12	MV-22 Osprey Tilt rotor vertical/short takeoff and landing (VSTOL)	aviation
4	AH-1Z Super Cobra attack helicopters	aviation

Table 11: MEU Aviation Characteristics for Support of A DO Platoon [3], [21], [92]

Characteristic	MV-22	CH53E	AH1-Z
Crew	2 pilots	2 pilots & crew chiefs	2 pilots
Capacity	24 troops (seated), 32 troops (floor loaded)	37 troops (55 with centerline seats)	6,661 lb (3,021 kg)
Payload	15,000 lbs external 20,000 lbs internal	32,000 lb (15,000 kg)	NA
Empty weight	33,140 lb (15,032 kg)	23,628 lb (10,740 kg)	12,300 lb (5,580 kg)
Loaded weight	47,500 lb (21,500 kg)	33,500 lb (15,227 kg)	18064 lb (8194 kg)
Max takeoff weight	60,500 lb (27,400 kg)	42,000 lb (19,100 kg)	18,500 lb (8,390 kg)

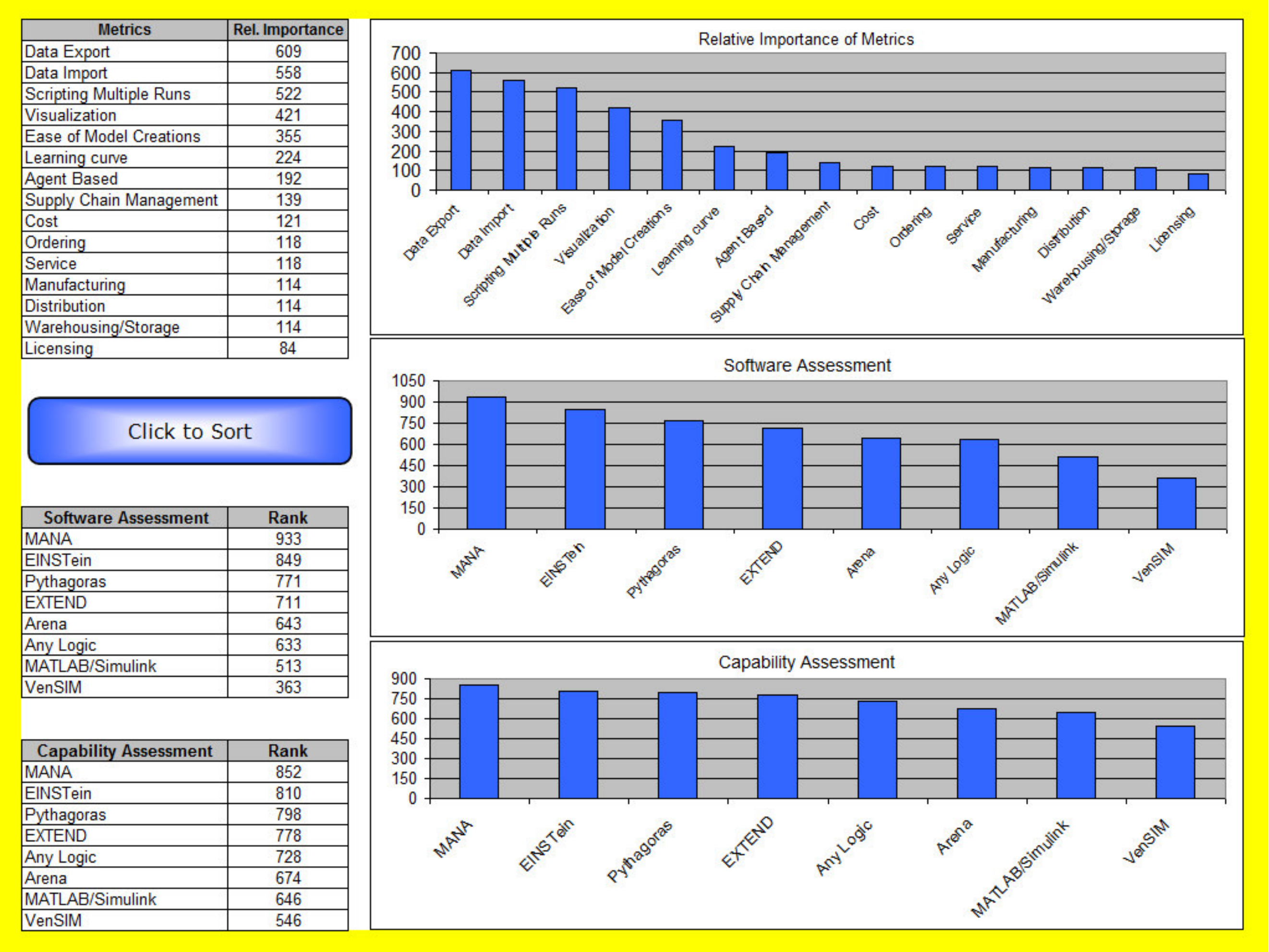
Table 11: MEU Aviation Characteristics Continued

Characteristic	MV-22	CH53E	AH1-Z
Maximum speed	275 knots (316 mph, 509 km/h)	170 knots (196 mph, 315 km/h)	160 knots (184 mph, 296 km/h); Never exceed: 222 knots (255 mph, 411 km/h)
Cruise speed	214 knots (246 mph, 396 km/h) at sea level	150 knots (173 mph, 278 km/h)	134 knots (150 mph, 248 km/h)
Range	879 nmi (1,011 mi, 1,627 km)	540 nmi (1,000 km)	370 nmi (426 mi, 685 km)
Combat radius	370 nmi (430 mi, 690 km)	240 nmi (300 mi, 400 km)	125 nmi (144 mi, 231 km) with 2,500 lb (1,130 kg) payload
Ferry range	2,417 nm (2,781 mi, 4,476 km)	990 nmi (1,139 mi, 1,833 km)	NA
Service ceiling	26,000 ft (7,925 m)	18,500 ft (5,640 m)	20,000+ ft (6,100+ m)

5.3.3 Step 3: Create the Modeling and Simulation (M&S) Environment

The creation of a M&S environment was accomplished through the use of an ABM software package called MANA. Selection of this tool for M&S was determined by conducting an across the board comparison to a number of available military ABM applications that the author had available to him. A custom Software Selection Tool (SST) was developed by the author and was utilized for this effort. Some of the M&S tools were discussed in Section 2.14 and the others applications considered are discussed in Appendix A. This tool utilizes similar functions performed in a Quality Functional Deployment (QFD) that is used to align a user's needs and requirements characteristics. This process ultimately aids in determining driving factors that can be used to guide a decision for the user. Although the author has only discussed the MANA scenario in detail thus far, there were several other scenarios available in Pythagoras and EINSTein that the author received from either the NPS or contacts within the military that were considered as viable for use and application via the ATLAS method. The result of the

SST was that MANA was the highest ranked tool with the results shown in Figure 38. Details of how the SST works can be seen in Appendix A. As can be seen the outcome



Metrics	Rel. Importance
Data Export	609
Data Import	558
Scripting Multiple Runs	522
Visualization	421
Ease of Model Creations	355
Learning curve	224
Agent Based	192
Supply Chain Management	139
Cost	121
Ordering	118
Service	118
Manufacturing	114
Distribution	114
Warehousing/Storage	114
Licensing	84

Software Assessment	Rank
MANA	933
EINSTein	849
Pythagoras	771
EXTEND	711
Arena	643
Any Logic	633
MATLAB/Simulink	513
VenSIM	363

Capability Assessment	Rank
MANA	852
EINSTein	810
Pythagoras	798
EXTEND	778
Any Logic	728
Arena	674
MATLAB/Simulink	646
VenSIM	546

Figure 38: Final Results Sheet of SST

had a close ranking for MANA, Pythagoras, EINSTein, and EXTEND. Ultimately, any of these four ABM&S applications could have been used to conduct this analysis.

The scenario used by CPT Bain and described in his Masters Thesis [8] is that of a USMC distributed operation platoon defending a sector of land from enemy combatants. This platoon is composed of three squads and two command elements. The general idea behind this force composition is to have a military unit that is highly adaptable and robust to meet the demanding new operational environments that range from humanitarian aid missions to full combat operations. Integral to this platoon's support is the ability to quickly request and receive logistical support from the Marine Air Ground Task Force (MAGTF) commander and the supporting elements under his purview. It is assumed a platoon of this nature will receive its primary logistical support from a seabased environment.

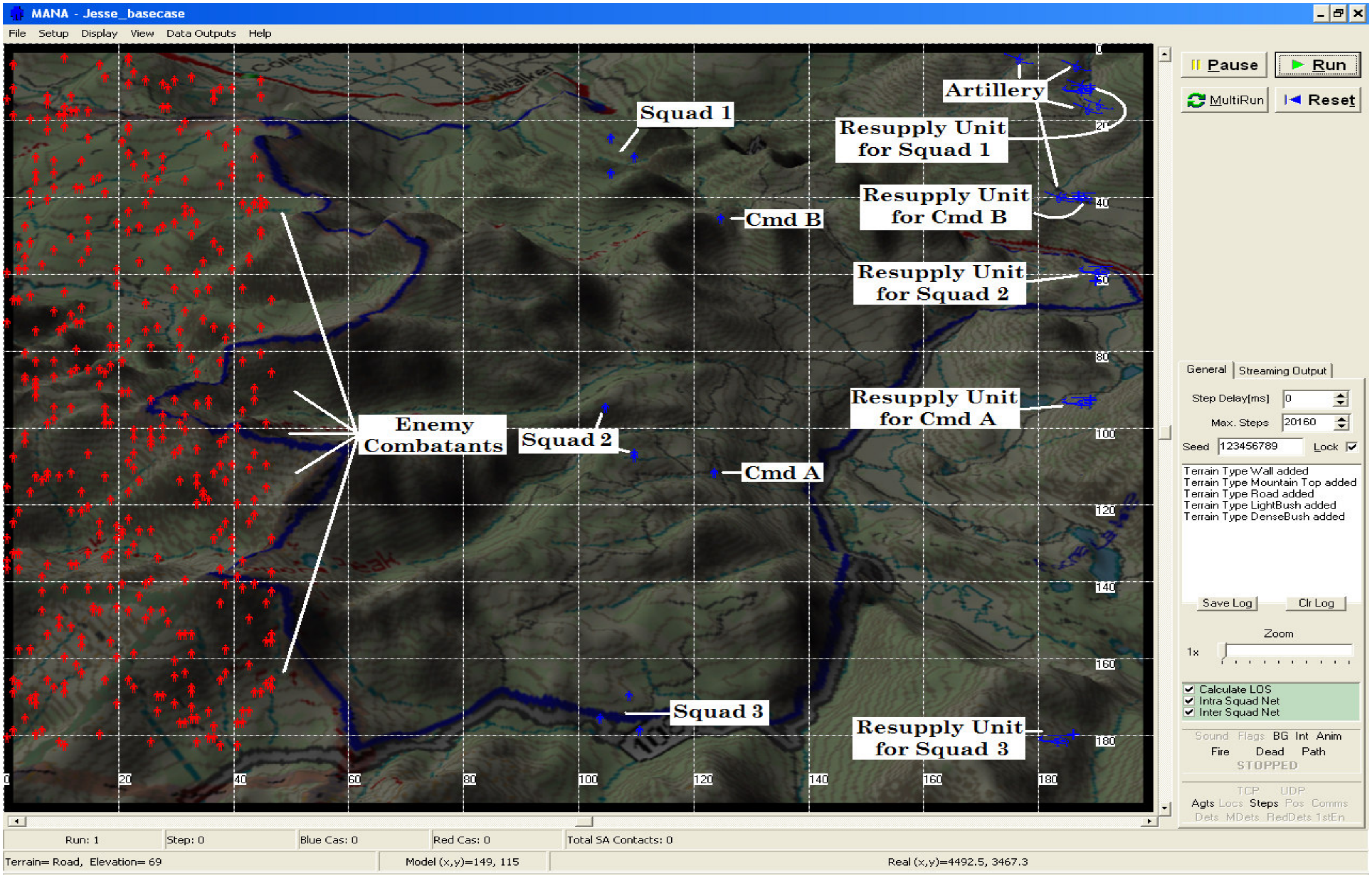
Although a great start, this model did have its drawbacks. A review of the model found that for the time it was created the model sufficed to provide the answers CPT Bain was looking for: to determine the optimum number of days (based on standard planning factors) of supply a platoon needed to be able to initially carry into combat. Ultimately this answer was three days. But what the author discovered as he delved deeper into the model was an understanding that fidelity needed to be increased and adjustments to the agent characteristics needed to be made. Doing this would provide a clearer picture into not only what made sense for the platoon to carry with it into a mission, but also provide insight into the underlying logistical support architecture required to support the DO platoon. As the author has pointed out, standard planning factors work well when aggregates are used because the principles of the Central Limit Theorem begin to appear, but for small forces that will be required to dynamically change to meet the mission need, these planning factors perform poorly as was shown in Section 5.2.1. These traditional planning factors have been developed based on large scale combat operations and distributed over a large number of troops involved. Consequently, they can not and should not be assumed to be viable for asymmetrical, nonlinear, distributed operations the military expects to be involved in for the future. Several volleys of correspondence between CPT Bain and the author resulted in a better understanding of the model's foundations, but the author also realized that the model in its current form would need some modification. To accomplish this, the author also looked at different ways to model the scenario more realistically in MANA so that logical real world outcomes would be the result. This will be covered in the next section.

5.3.3.1 MANA

MANA is capable of conducting singular or multiple runs of the same input conditions for a variety of any of the modifiable variables the author wishes to define. By performing this task the user will see slightly different outcomes since MANA is an ABM&S application that automatically assigns a random variable seed to slightly perturb the model each time it is run. This facilitates an agent's ability to learn, act, and react based on user specified personality traits. From these runs the user can access results of the simulation via various data output reports. These reports are generated at the end of each run and are in the form of comma separated value files. Alternatively MANA is capable of running a series of different experiments or DOE autonomously via CONDOR, a program that accesses unused computing power of computers on a local area network. This is made possible through a two specific programs that were developed at the NPS called *XStudy* along with another program, *Old McData* that bridges MANA with CONDOR to run the DOE. The use of these tools significantly reduces the run time of the experiments by executing MANA's code without visualization through the MANA GUI. These two programs will be covered after the MANA section.

The first task facing the author was to modify the model so that it would more accurately represent the logistical support structure for the platoon at an entity (squad or command group) level. The layout of the model did not necessarily need to reflect reality, but the results generated by the model would need to reflect reality. One of the major changes to Bain's model consisted of modifying the resupply component of the model. Run observations of the original model showed that upon a resupply request from one entity, whether it was a rapid request or scheduled mission, resupply helicopters were sent out to all five entities (three squads and two command units) whether they needed it or not.

For example, if squad one was experiencing heavy enemy contact, thus using more of its resources than say squad three, who was experiencing very little enemy contact, a resupply request would be sent over the communication network that squad one was in need of resupply. This order was then processed and then all resupply helicopters would deploy and resupply the entire platoon back to the initial days of supply they left with (i.e. if they left with three days of supply the resupply mission results in a replenishment of each entity back to a full three days of supplies). Inherently one can see there is a problem in accounting for actual logistical needs in the form of consumables and aviation asset usage in this fashion. In the real world, these actions would result in squad three ending up with a small cache of supplies if squad one continued to experience a high level of enemy contact and result in a significant amount of unnecessary resupply flights to the other squads. To resolve this issue each entity was provided a singular line of communication to a sensor that relayed request signals to a dedicated resupply agent solely for that entity. This modification should provide an accurate accounting of resupply runs conducted to each of the five singular DO platoon entities in the model. Another modification made to the model was that in the real world as supplies get low, a request for resupply by the platoon would likely occur via communication links. In the original model an agent/entity only sent a resupply request when it ran out of supplies, represented by its “fuel tank” going to zero. This resulted in the agent going into a dormant state, thus allowing enemy agents to pass unchallenged until the agent was resupplied by aerial assets and could engage the enemy agents again. To overcome this drawback the resupply requests were modified to communicate a need prior to an agent’s fuel tank reaching zero. A snap shot of the model in its basic starting condition in MANA with all agents (enemy combatants, three squads, two command groups, resupply helicopters, sensors, and the artillery battery) can be seen in Figure 39. This is best viewed in color.



Step Delay[ms] 0

Max. Steps 20160

Seed 123456789 Lock ☑

```
Terrain Type Wall added
Terrain Type Mountain Top added
Terrain Type Road added
Terrain Type LightBush added
Terrain Type DenseBush added
```

☑ Calculate LOS
☑ Intra Squad Net
☑ Inter Squad Net

Run: 1 | Step: 0 | Blue Cas: 0 | Red Cas: 0 | Total SA Contacts: 0

Terrain= Road, Elevation= 69 | Model (x,y)=149, 115 | Real (x,y)=4492.5, 3467.3

Figure 39: Screen shot of DO Platoon Model in MANA

The variables that drive the primary agent characteristics in the model consist of 14 input variables. Ten of these variables consist of parameters that can either be quantified and or controlled by the logistician or decision maker. The remaining four variables are noise variables that are out of control of the decision maker and must be estimated based on intelligence means. These factors along with a set of ranges were utilized to create a DOE to conduct scenario investigations for a varying number of operational conditions. The list of variables, ranges, and their definitions are in Table 12.

Table 12: Table of Design Variables for DO Platoon Model in MANA

Variable	Range	Units	Description
Controllable Factors			
X1: DO Squads Days of Supply	1 – 5	Days	Amount of supplies carried into initial deployment.
X2: Scheduled Resupply Rate	1 – 5	Days	Number of days between scheduled resupply missions
X3: Sense and Respond Lead Time	0 – 12	Hours	Amount of time it takes for the autonomous logistics system to respond to a request
X4: Consumption Rate when in Enemy Contact	1 – 50	# / time unit	Consumption rate of the DO squads when in contact with the enemy
X5: Consumption Rate when Shot at	1 – 50	# / time unit	Consumption rate of the DO squads when in direct fire contact with the enemy
X6: Rapid Request Setup Time	1 – 6	Hours	Time from resupply request to departure of helicopter from logistic support area
X7: Time to Unload Resupply Helicopter	1 – 60	Min	Time to execute transfer of supplies and casualties from helicopter to the agent
X8: Resupply Speed	50 – 1000	m/s	Speed of resupply helicopter
X9: Resupply Stealth	0 – 100	%	Concealment from enemy
X10: Friendly Inorganic Sensor Persistence	1 – 60	Min	Tine a friendly force sensor will maintain history of a report of enemy contact
Uncontrollable (Noise) Factors			
X11: Enemy Sensor Range	0 – 2000	M	Enemy sensor range of DO agent presence
X12: Enemy Squad Size	1 – 14	#	Number of enemy agents in a squad
X13: Contact Persistence Enemy	1 – 60	Min	Time that an enemy agent will keep track of a DO agent
X14: Hits to Kill Enemy	1 – 10	#	Hits to kill an enemy agent

Within the MANA model exists several personality settings for the agents. Some of the settings remained unchanged, while others were modified by the author to account for changes made to the model or reflect the modified resupply characteristics of the model. The intent was to keep the model as original as possible since it was already vetted by military advisors and modelers who aided CPT Bain in his work and validated the model. This approach will help in the accomplishment of Step 4. Unfortunately the author did need to make some changes to reflect the added investigation into resupply characteristics of the model. The original model it was found also did not differentiate between friendly agents that a resupply agent (helicopters) would ultimately resupply. Thus in the original model a helicopter by happenstance would resupply any agent that it was situationally aware of through its sensor map. To help eliminate this "accidental" resupply occurrence, careful selection and placement of resupply routes were articulated in the model and the awareness radius for the resupply agent was reduced to a very small coverage area.

Because MANA allows for a random seed to be utilized for analysis of the ABM&S environment, each of the 1000 DOE cases was to be run five times to get a distribution of the outcome for that particular set of initial conditions. The DOE was generated according to the ranges in Table 12 with a Latin Hypercube Sampling (LHS) function in MatLab. The function used to create the DOE, *lhsdesign* is as follows: *X = lhsdesign(1000,14,'iterations',10, 'criterion','correlation').* This function generates a LH sample or table with numbers between zero and one, *X*, containing **n** or *1000* rows of **p** or *14* variables. The values are randomly distributed with one from each interval (0,1/n), (1/n,2/n), ..., (1-1/n,1), and are randomly permuted [77]. By specifying the criterion *correlation*, the LHS design was iteratively generated to reduce the correlation of the points and this was attempted at most *10* times to arrive at the best design. Of the many DOE options available the LHS was selected for it ability and good results when performing uncertainty in computer model outputs based on the uncertainty of the inputs to the model [84]. For an in depth description of how this applied the reader is referred to [84] by McKay who gives a good overview of the use of the Latin Hypercube and its applicability in this fashion. The DOE generated by MatLab was placed in the statistical

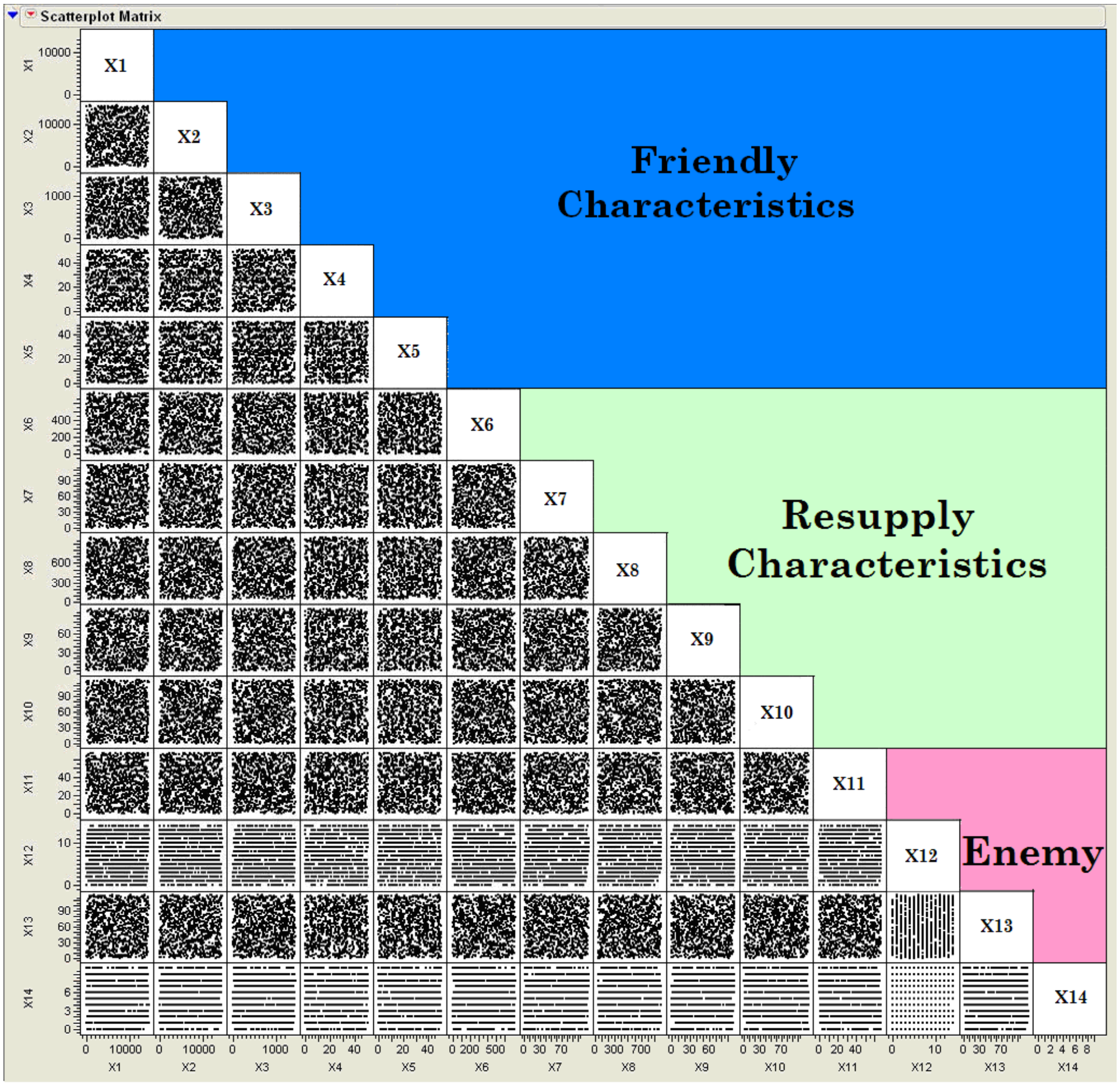


Figure 40: Scatterplot Matrix for LHS Design – 1000 Cases

software JMP to evaluate the coverage of the design space input ranges. The scatter plot matrix of the design points generated is seen in Figure 40.

Further analysis of the designs generated revealed by inspection that 53 of the designs were infeasible. This was determined by observing scenario inputs that were unlikely to happen in the real world. An example of this would be a DO platoon leaving with only one days worth of supplies and not incurring rescheduling activities until the third day. Several of these cases were run and the output generated either zero percent

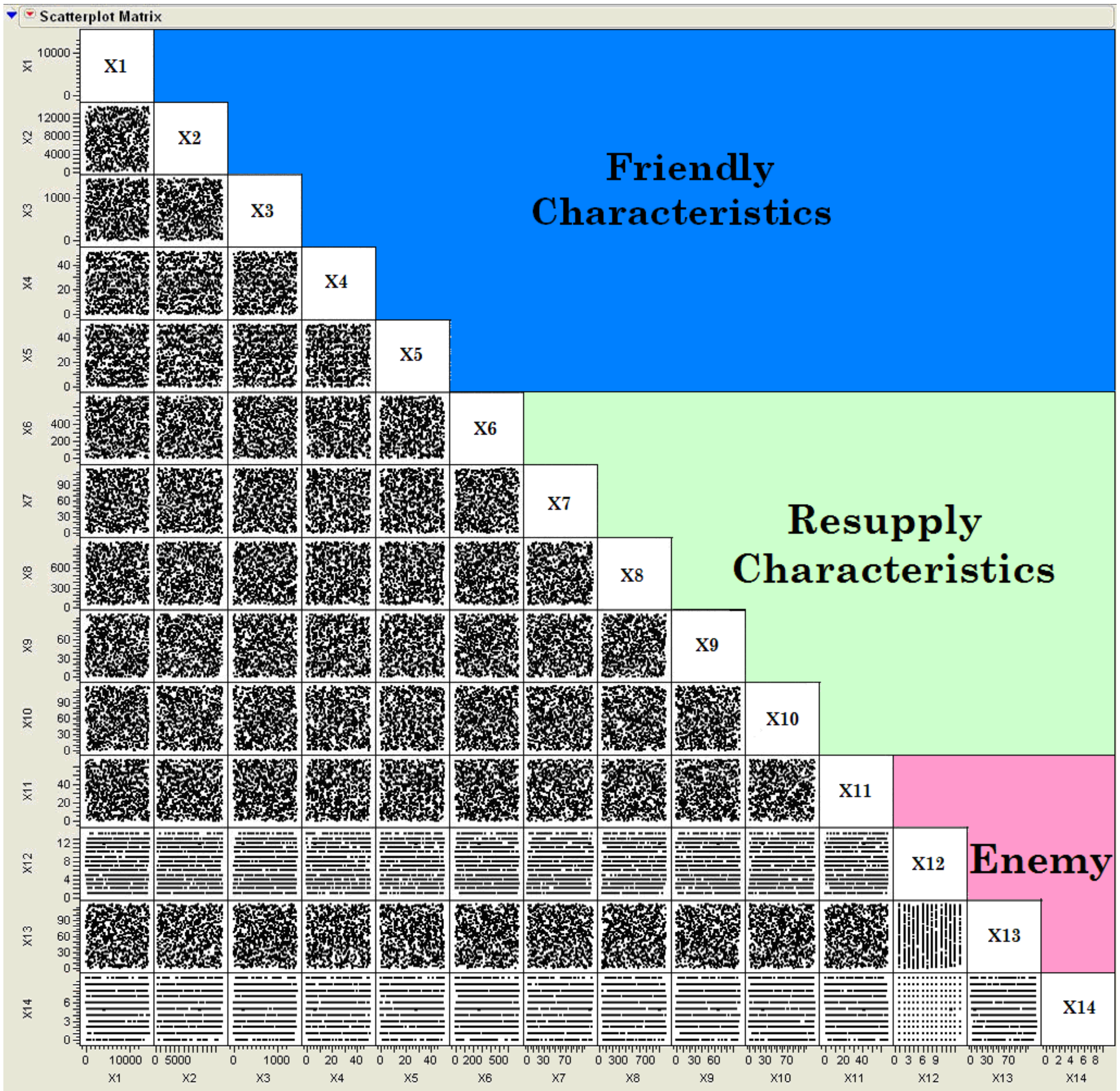


Figure 41: Scatterplot Matrix for Filtered LHS Design - 947 Cases

enemy killed or almost continuous helicopter resupply operations as the ground agents were constantly sending rapid requests to their respective sensors that resupply of their resources was required. Upon completion of this vetting process, 947 cases remained and they too were placed in JMP to visualize the designs coverage of the design space. This can be seen in Figure 41.

As can be seen, the comparison between Figure 40 and Figure 41 shows very little impact to the coverage of the input variable design space. With the DOE generated the next step is to set up the MANA model to execute the DOE. This is accomplished by interfacing MANA with two programs, *XStudy* and *Old McData*, generated by the NPS for design space exploration or data farming as they refer to it via MANA and its output reports. These two applications will be discussed briefly next.

5.3.3.2 XStudy

This application is a GUI based program created specifically for generating a study file in the XML format. This file specifies how a user wants to conduct an automated multi-simulation experiment in MANA. Through the *XStudy* GUI the user can specify variables to be varied associated with a particular scenario file generated in an XML format, as MANA's is. Through *XStudy*, the user has the option of specifying how the prescribed parameters will be varied, either by a full factorial DOE created by *XStudy* or the user can use a custom experimental design specified by a .CSV file. The LHS DOE described earlier was placed in Microsoft Excel and manipulated to associate the LHS DOE generated in MatLab to that of the 14 variables in Table 12. The author was then able to specify the *XPaths* or directional path for these variables, essentially defining the parameters to be perturbed in MANA as each DOE case is executed when interfacing with the MANA software application. Once all the *XPaths* are specified, the user clicks the "Generate Scenario" button and *XStudy* generates a *study.xml* file. This file is then used by *Old McData* to execute the runs autonomously. A partial sample of the authors *XStudy* files can be seen in Appendix B 2. With the *XStudy* file generated, the author could now use *Old McData* to execute the DOE in MANA.

5.3.3.3 Old McData

Old McData (OMD) gets its name because NPS used this program to data farm or explore the results of various ABM&S applications, primarily MANA and Pythagoras. It is designed to allow the user to farm for specific data from large experiments to multiple replications of a single design run on one computer. OMD performs this operation by interfacing with *CONDOR*, discussed earlier, to execute multiple runs instantiated by the DOE. OMD allows for four types of data gathering, they are [137]:

1. Gridded Data
2. Cartesian Product Generation
3. Specifying Runs Desired via a .CSV file
4. Evolutionary Programming Algorithms (i.e. Genetic Algorithms & Simulated Annealing)

For the purposes of this research, the third option was used. OMD once installed and coded to execute the created study file and MANA file is a relatively easy task of just typing in command executables. Although this sounds simple, much work was needed to debug and get the code up to this step. The author would be remiss if he did not include and thank Mr. Steven Upton, the creator of *Old McData* and *XStudy*, who was a great help in correcting and guiding the author in fixing errors in several files, answering questions, and spending his own time examining troublesome code in both of these programs.

Upon completion of the debugging process, the author executed several runs in OMD to get an idea of report outputs and how to best utilize the varying categories of data in several standardized reports available as output from MANA. The result of this exploration will be covered in Step 5.

5.3.4 Step 4: Verify, Validate, Accreditation the M&S Environment

In this context the MANA model has already been validated and accredited because the scenario's foundations and creation have been assessed, tested, and been accepted by the military community as a valid model of the notional environment. Verification needs to be performed to assess the changes made to the original model. MANA has several physical aspects, like terrain that can be modeled within its framework and can be seen in Table 13. Each terrain type has its own unique set of parameter values for Going, Cover, and Concealment [82].

- Going defines how the terrain affects an agent's speed and values range from 0.0 to 1.0. A value of 1.0 means the agent can move at its normally defined speed. A value of 0.5 would have the agent moving at half speed and a value of 0.0 would prevent the agent from moving at all.
- Cover defines the degree to which agents can be shot by direct fire weapons in the terrain. A value of 0.0 means the agent has no cover whatsoever and can be shot as if they were in open terrain. A value 1.0 means the terrain provides full protection from weapons fire.
- Concealment defines the degree to which an agent can be seen in the terrain. A value of 0.0 means the agent is fully visible while a value of 1.0 means the agent is completely concealed.

Table 13: MANA Physical Terrain Parameters [82]

Terrain Feature	Description
Billiard Table	Colored black, plain terrain that has no special properties.
Easy Going	Terrain that represents a road or other region that is particularly attractive to move along. Yellow coloring represents these areas. Entities can have personality weightings set towards Easy Going terrain. Thus, a “convoy” can be made to stay close to a road if the surrounding terrain would affect its movement speed.
Wall	Obstacle terrain that is represented by light grey coloring. No entity may occupy an obstacle cell. Entities can see through wall cells only if ‘Line of Sight’ is turned off, and entities can fire kinetic energy weapons through wall cells only if the weapons are explicitly set to allow this.
Light Bush/Dense Bush	Bush terrain is represented by green coloring. Differing density provides different movement speed, cover from weapons fire and concealment multipliers.
Hilltop	Hilltop terrain is represented by dark grey coloring. A high level of concealment makes this terrain ideal for launching indirect weapons.

The agents themselves have a wide range of tangible characteristics that can be modeled such as the agents speed or mobility, color that can be used to represent agent allegiance (friendly, enemy or neutral) or state the agent is in, the number of shots to kill an agent or armor thickness for vehicles. Other personality settings defined by the user also add realistic characteristics to the agents modeled. For example a distance penalty function is used by the agent to determine which direction to move to for the next time step based on obstacles, enemy awareness, etc. Further definitions of the available tangible agent characteristics can be seen in [82] Section 3.10 for the interested reader. *Tangible agent characteristics* are defined as things that can be physically modeled according to doctrine or how a soldier would react when posed with information about their environment.

What is difficult to model are the intangible aspects of the combat environment. In the model these aspects are how would agents react, interact, or perform certain functions based on learning as well as other functions that humans in the real world execute on a situationally dependant basis. Within MANA there are several agents that have to be modeled for the DO scenario and they along with their description are listed in Table 14.

Table 14: Agent Modeled in MANA DO Platoon Scenario

Agent (Qty)	Description
DO Squad (3)	These are the three squads that will perform the bulk of the mission and come into contact with enemy combatants.
Command A (1)	This the command element lead by the Platoon Leader of the DO Platoon. This agent communicates with the entire platoon and is directly responsible for C^2 of the DO platoon.
Command B (1)	This the command element lead by the Platoon Sergeant of the DO Platoon. This agent also communicates with all three squads and the Command A element and is directly responsible for command and control of the DO platoon in the event the Command A element is rendered ineffective.
Logistics Sensor (5)	This sensor is in direct communication with one of the squads or command elements and represents the resupply coordinator who directs the combat service support area.
Resupply Helicopter (5)	This agent represents the aerial resupply asset arriving from a location that is 50 nm away.
Artillery Fire Support (6)	This agent communicates with all squads and command elements to coordinate indirect fires on enemy combatants.
Enemy Combatant (40)	These agents represent the enemy agents that the DO squads will attempt to eliminate. They can range from 1-14 in size resulting in at least 40 or up to 560 combatants per the DOE.

Within the model certain personality traits and trigger states were adjusted to fit the incorporation of the modified resupply agents, but in most cases the agent triggers that dictated agent state changes, like going from a patrol status to combat status were left alone or modified to act more stringently. When a more stringent change was imposed, the author felt the model portrayed more of an optimistic view of agent traits and capabilities.

Once the model was finalized several test cases were performed and the output results were analyzed to see if the modified model was performing as intended. This observational analysis of the model outputs completes this step by verifying that the model works as intended. To quantify this the author compared the only MOE *Percent Enemy Killed*, originally looked at by Bain, to the authors results and found no significant order of magnitude difference in the outcome for this MOE. This task was accomplished by reviewing data results received from the NPS for runs conducted by CPT Bain as well as looking at Bain's thesis work. The author was not able to verify the additional MOE, *Number of Resupply Helicopters*, added to the model because the report used to calculate this output was not part of the data package Bain used for his work. Even if it had been included the results would not have been useful as the model needed t be modified to more accurately represent realistic resupply activities based on individual entity requests.

5.3.5 Step 5: Create / Update a Case Based Database

Creation of the database from the MANA runs that would be used to conduct CBR methods involved understanding the outputs of nine reports that MANA is capable of producing. Unfortunately in MANA these reports are standardized and cannot be easily modified to narrow or scope the information output. This fact reinforced a statement CPT Bain made to the author through email correspondence of how MANA was a round peg in a square hole for his thesis. While MANA served a means to an end for his work, it was difficult for CPT Bain to manipulate MANA outputs to get particular data he would have ideally liked to get from MANA to perform his analysis. The MOE of *Percent Enemy Killed* although not a measure of logistics is directly correlated to how much *stuff* the squad would need to accomplish its mission. A high rate of success in this MOE should be directly correlated to having sufficient resources to do the mission and represent a run in which supply support was adequate. Therefore, this MOE was retained by the author as a useful output for the analysis.

Test runs that generated all of the reports available resulted in the author selecting two specific reports. The first was the *Agent State Data* (ASD) which is taken at the end of the simulation and the *Position Data* (PD) report that records the *x*, *y* coordinate position for each agent at every time increment of the simulation. The ASD file contains the agent id, agent name, the squad and sub-squad of the agent, the agent's *x*, *y* position and status (active, injured or dead) at the end, the *x*, *y* position of the agent's goal, the number of hits the agent received, name of current state, step at which the current state was entered, and finally the agents current "fuel" available. For Bain's work, this file provided the necessary total information of enemy casualties and those still alive, thus the percentage of enemy killed could be determined.

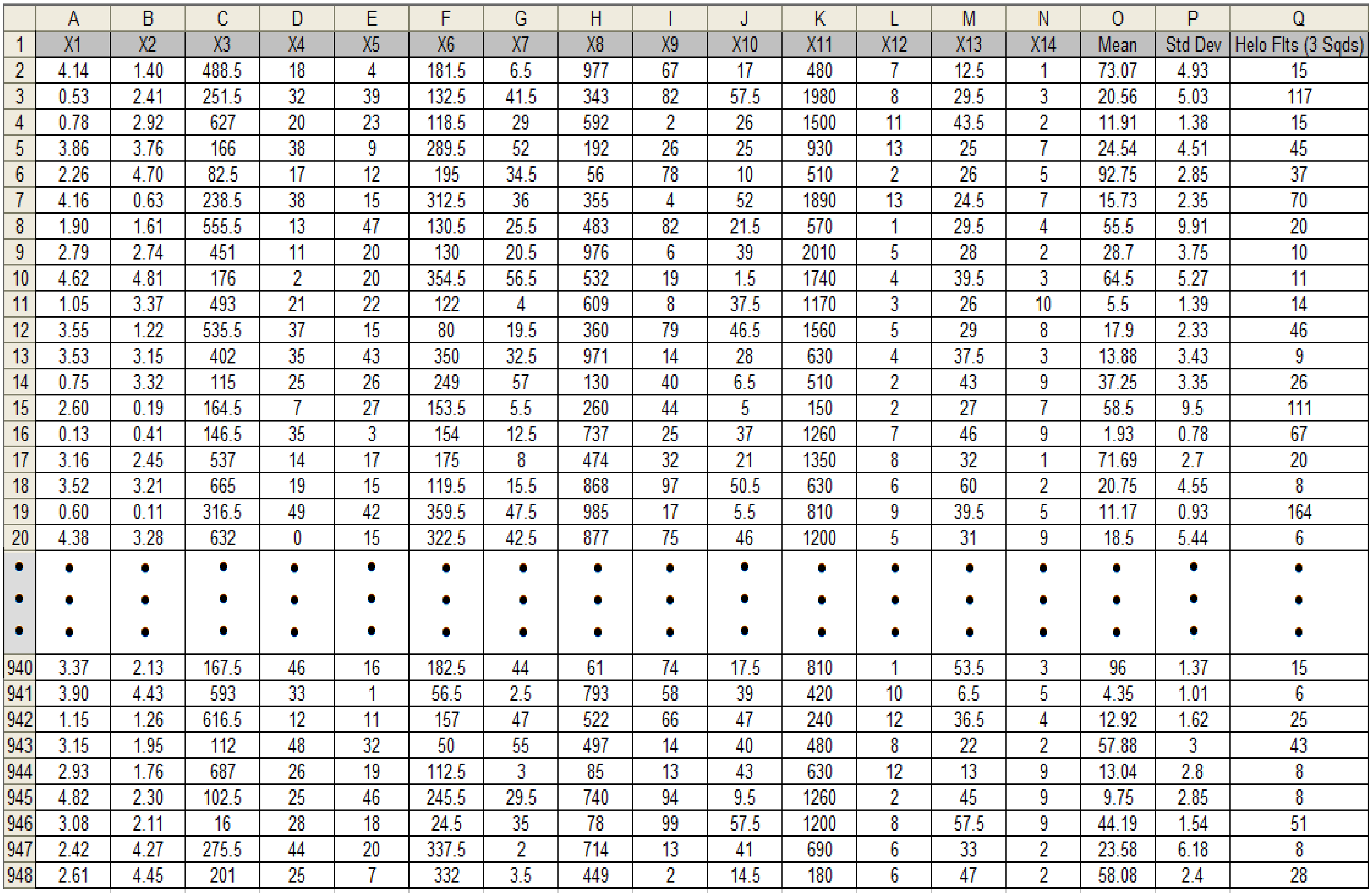
To determine the logistical need for the DO platoon another report needed to be utilized. This is where the PD file was utilized. The PD file displays the model time step, agent identification number, agent status (recorded as 2 for active (normal state), 1 for injured, and 0 for dead.), and the agent (*x*, *y*) position. The last data point, the agent (*x*, *y*) position, is what the author decided to utilize to determine the logistics delivery schedule and frequency for the DO platoon agents. By utilizing the author's defined *(x, y)* coordinate position specified in the model for the landing point for the resupply helicopter an accounting of frequency and when the helicopters arrived could be determined. Since the helicopter was set up to only be at the landing point for one time step and each time step was recorded, the PD file theoretically via a filtering algorithm could be used to go through the file and extract the relevant data of time on the landing point for the resupply agent. This would allow the author to determine the time at which the resupply helicopter reached its landing point, for what squad, and over the course of the entire scenario the total number of flights conducted, either cumulatively or for each friendly element.

Initial exploration of the runs performed and the resulting two reports by the DOE found a large range of file size for the reports necessary to calculate the MOE's. The files ranged as small as a few kilobytes to as large as 222 MB. Immediately memory allocation was identified as an issue and the need for filtering codes to not only get the data, but also to reduce the amount of superfluous information was realized. To solve the data issue 200 GB of temporary additional storage capacity was obtained on a server and a 100 GB external hard drive was acquired. For the data extraction several post processing filtering algorithms were written in MatLab and were utilized to significantly reduce the size of most reports to less than 30 kilobytes. The post processing code created can be seen in Appendix C 1.

In the end, the execution of the 947 case DOE, executed 5 times equated to 4735 cases and required just under the 300 GB of computer drive space for the two type reports selected. This was a limiting factor in executing more replications of the DOE cases for the purpose of determining output distributions on the data. The experiment itself took approximately two weeks of 24 hours a day running via CONDOR at the Aerospace Systems Design Lab (ASDL). At any one point as little as a few computers to as much as 95 were being utilized to execute the MANA runs and generate the output reports. Considering that CPT Bain only had 12 computers at his disposal and it took about a month to run his experiments, this was relatively fast. Again the limiting factor for the author in conducting more experiments in MANA was the hard drive space needed to store the initial output reports from MANA.

Ultimately MatLab was used to consolidate and extract all of the data from the MANA reports to create a database in Excel consisting of the input variables from the DOE and the MOE's of *Percent Enemy Killed* and *Number of Resupply Flight* conducted. The use of readily accessible software applications for the CBR database like Excel is important for several reasons. It is accessible to military personnel at all levels (as long as they have a computer), it easily accessed for updates to the database as additional data is collected, and the data is easily transferrable. A sample of the database is shown in Table 15.

Table 15: Sample of DO Platoon Operational Database



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	Mean	Std Dev	Helo Flts (3 Sqds)
2	4.14	1.40	488.5	18	4	181.5	6.5	977	67	17	480	7	12.5	1	73.07	4.93	15
3	0.53	2.41	251.5	32	39	132.5	41.5	343	82	57.5	1980	8	29.5	3	20.56	5.03	117
4	0.78	2.92	627	20	23	118.5	29	592	2	26	1500	11	43.5	2	11.91	1.38	15
5	3.86	3.76	166	38	9	289.5	52	192	26	25	930	13	25	7	24.54	4.51	45
6	2.26	4.70	82.5	17	12	195	34.5	56	78	10	510	2	26	5	92.75	2.85	37
7	4.16	0.63	238.5	38	15	312.5	36	355	4	52	1890	13	24.5	7	15.73	2.35	70
8	1.90	1.61	555.5	13	47	130.5	25.5	483	82	21.5	570	1	29.5	4	55.5	9.91	20
9	2.79	2.74	451	11	20	130	20.5	976	6	39	2010	5	28	2	28.7	3.75	10
10	4.62	4.81	176	2	20	354.5	56.5	532	19	1.5	1740	4	39.5	3	64.5	5.27	11
11	1.05	3.37	493	21	22	122	4	609	8	37.5	1170	3	26	10	5.5	1.39	14
12	3.55	1.22	535.5	37	15	80	19.5	360	79	46.5	1560	5	29	8	17.9	2.33	46
13	3.53	3.15	402	35	43	350	32.5	971	14	28	630	4	37.5	3	13.88	3.43	9
14	0.75	3.32	115	25	26	249	57	130	40	6.5	510	2	43	9	37.25	3.35	26
15	2.60	0.19	164.5	7	27	153.5	5.5	260	44	5	150	2	27	7	58.5	9.5	111
16	0.13	0.41	146.5	35	3	154	12.5	737	25	37	1260	7	46	9	1.93	0.78	67
17	3.16	2.45	537	14	17	175	8	474	32	21	1350	8	32	1	71.69	2.7	20
18	3.52	3.21	665	19	15	119.5	15.5	868	97	50.5	630	6	60	2	20.75	4.55	8
19	0.60	0.11	316.5	49	42	359.5	47.5	985	17	5.5	810	9	39.5	5	11.17	0.93	164
20	4.38	3.28	632	0	15	322.5	42.5	877	75	46	1200	5	31	9	18.5	5.44	6
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
940	3.37	2.13	167.5	46	16	182.5	44	61	74	17.5	810	1	53.5	3	96	1.37	15
941	3.90	4.43	593	33	1	56.5	2.5	793	58	39	420	10	6.5	5	4.35	1.01	6
942	1.15	1.26	616.5	12	11	157	47	522	66	47	240	12	36.5	4	12.92	1.62	25
943	3.15	1.95	112	48	32	50	55	497	14	40	480	8	22	2	57.88	3	43
944	2.93	1.76	687	26	19	112.5	3	85	13	43	630	12	13	9	13.04	2.8	8
945	4.82	2.30	102.5	25	46	245.5	29.5	740	94	9.5	1260	2	45	9	9.75	2.85	8
946	3.08	2.11	16	28	18	24.5	35	78	99	57.5	1200	8	57.5	9	44.19	1.54	51
947	2.42	4.27	275.5	44	20	337.5	2	714	13	41	690	6	33	2	23.58	6.18	8
948	2.61	4.45	201	25	7	332	3.5	449	2	14.5	180	6	47	2	58.08	2.4	28

5.3.6 Step 6: Identify and Incorporate Uncertainty in Information

Uncertainty is abound when it comes to military operations, the phrase “fog of war” was coined my Prussian military analyst and tactician General Carl Von Clausewitz who wrote [23]:

> "The great uncertainty of all data in war is a peculiar difficulty, because all action must, to a certain extent, be planned in a mere twilight, which in addition not infrequently — like the effect of a fog or moonshine — gives to things exaggerated dimensions and unnatural appearance."

Military logisticians and tacticians rely on many things when trying to assess a situation and determine the best COA to promote mission success. These consist of, but are not limited to human and electronic intelligence, socioeconomic conditions, and often times their own personal intuition. Unfortunately all of these sources contain some level of uncertainty and can be difficult to access in a timely fashion. Capturing this uncertainty can be difficult because quantifying it is not an exact science, there is uncertainty in the uncertainty, and often times knowledge or the level of knowledge known is described linguistically. To account for this aspect of the decisions support system, the author has employed the principles of fuzzy logic and fuzzy membership functions. This subject was discussed in Section 2.10 with additional information in Appendix D.3.

Application of fuzzy logic has increased in a wide variety of areas where modeling of human thought processes and the use of linguistics is prevalent. So much so, that there are a variety of programs which have incorporated fuzzy applications as a part of their capabilities. One such program is MatLab. It contains a *fuzzy logic toolbox* that allows the user to create fuzzy inference systems (FIS) under either the Mamdani model or the Takagi-Sugeno-Kang (TSK) model. The primary capabilities available to the user with this toolbox are [37]:

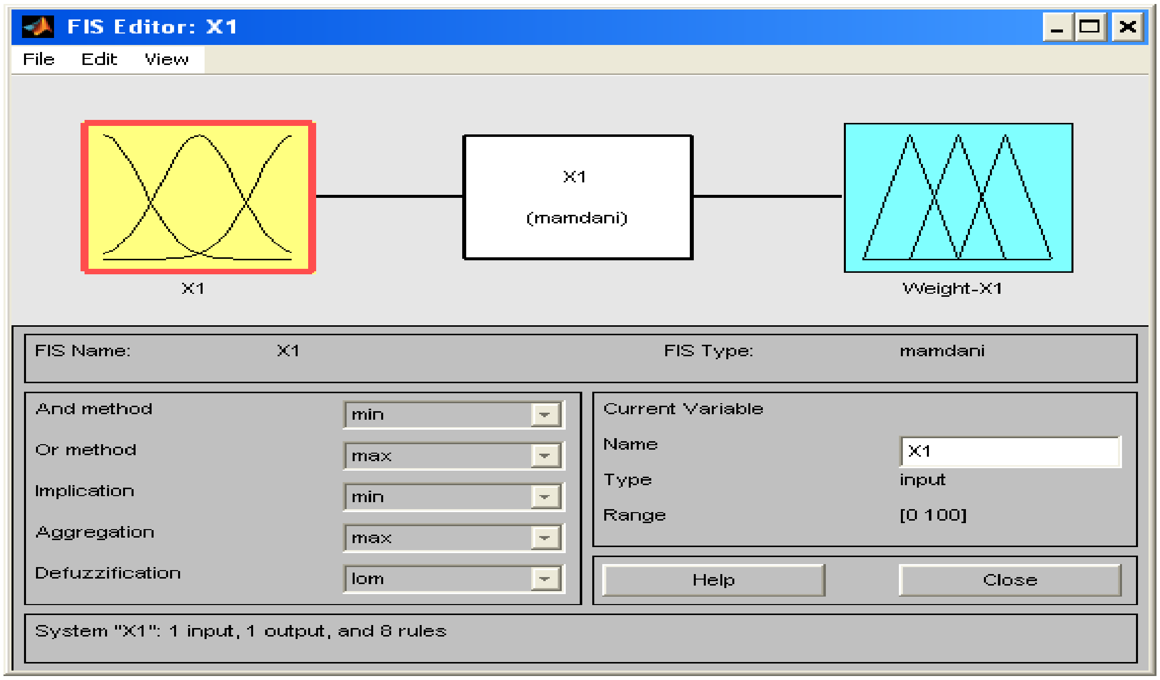
- Specialized GUIs for building fuzzy inference systems and viewing and analyzing results
- Membership functions for creating fuzzy inference systems
- Support for AND, OR, and NOT logic in user-defined rules
- Automated membership function shaping through neuro-adaptive and fuzzy clustering learning techniques

The Mamdani model applies linguistic fuzzy modeling for problems that focus on interpretability, while the TSK model applies precise fuzzy modeling for problems that are focused on accuracy. One of the main differences between these two models is that Mamdani model results in output membership functions that are fuzzy sets that will require defuzzification and the TSK model's output membership functions are either linear or constant. For these reasons and the environment that characterizes the current and future military environment, the Mamdani style FIS's were used in this step. For clarification, fuzzification is the process of converting a numerical value into a value based on a specified membership function and defuzzification is the process of converting a linguistic output variable into a numerical one that can be used for calculations or as a final result (refer to Section 2.10).

In this research the author provided a location in the user's GUI, described in Step 8, in which a level of confidence (0 – 100 %) in the data could be assigned for each of the 14 input variables. The first step in the process of utilizing this input is to fuzzify the input variable confidence assessment. This accomplished by defining input membership functions and rules with logical operators. The users input are evaluated against all the rules generated and the outputs are aggregated under a series of truncated output functions. The final step is to defuzzify the aggregated output function. Within MatLab there are five built-in methods supported for aggregation techniques; they are: centroid, bisector, middle of maximum (the average of the maximum value of the output set), largest of maximum, and smallest of maximum. The author used the centroid calculation, which returns the center of area under the curve. An example of this process in MatLab for the variable *X1* (Initial Days of Supply) is described below.

5.3.6.1 Creation of FIS for X1 (Initial Days of Supply)

The incorporation of the fuzzy logic toolbox in MatLab has enabled the creation of FIS with a lot more ease and flexibility than in the past. The task of creating membership functions, rules, and specifications is now accomplished through a series of GUI's. The following example outlines the process the author followed for the creation of a FIS for the variable X1. Typing the command *fuzzy* in the MatLab command window starts the process and what appears is the FIS editor as shown in Figure 42. This allows the user to specify the variable input and output names, what type of FIS (Mamdani or TSK), as well as a variety of methods for calculating the FIS output. Double clicking on the input (left or yellow box labeled X1) or output (right or blue box labeled Weight-X1) icons brings up the membership function editor. This GUI allows the user to specify the type of membership function to characterize the input and output variables. The available options consist of 11 types of shape functions, i.e. triangular, trapezoidal, Gaussian, etc. For X1, the variable is the amount of supplies that the DO Platoon will have on hand for their initial deployment. This knowledge should be relatively known and with some certainty so the author selected a normal / Gaussian membership function to represent the users input in how confident they are in the current



FIS Name: X1
FIS Type: mamdani

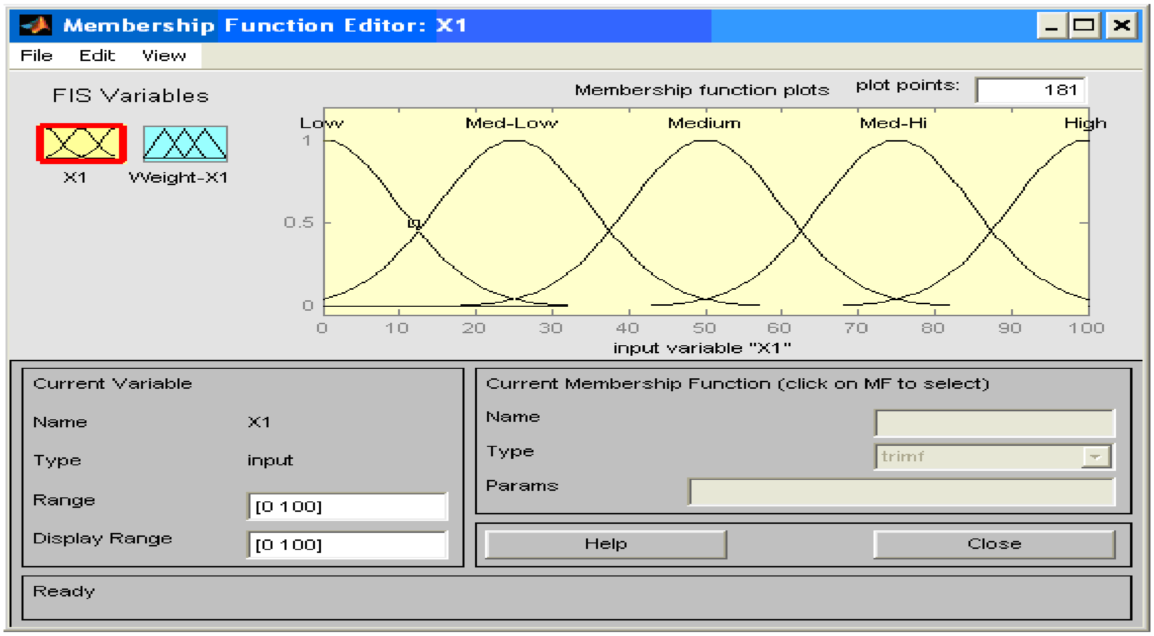
And method: min
Or method: max
Implication: min
Aggregation: max
Defuzzification: lom

Current Variable
Name: X1
Type: input
Range: [0 100]

System "X1": 1 input, 1 output, and 8 rules

Figure 42: FIS editor for X1

scenario information. The confidence scale is from 0-100 % for all of the variables of interest. X1 is broken down into five levels of linguistic classification of *Low, Med-Low, Medium, Med-High,* and *High*. This can be seen in Figure 43.



plot points: 181

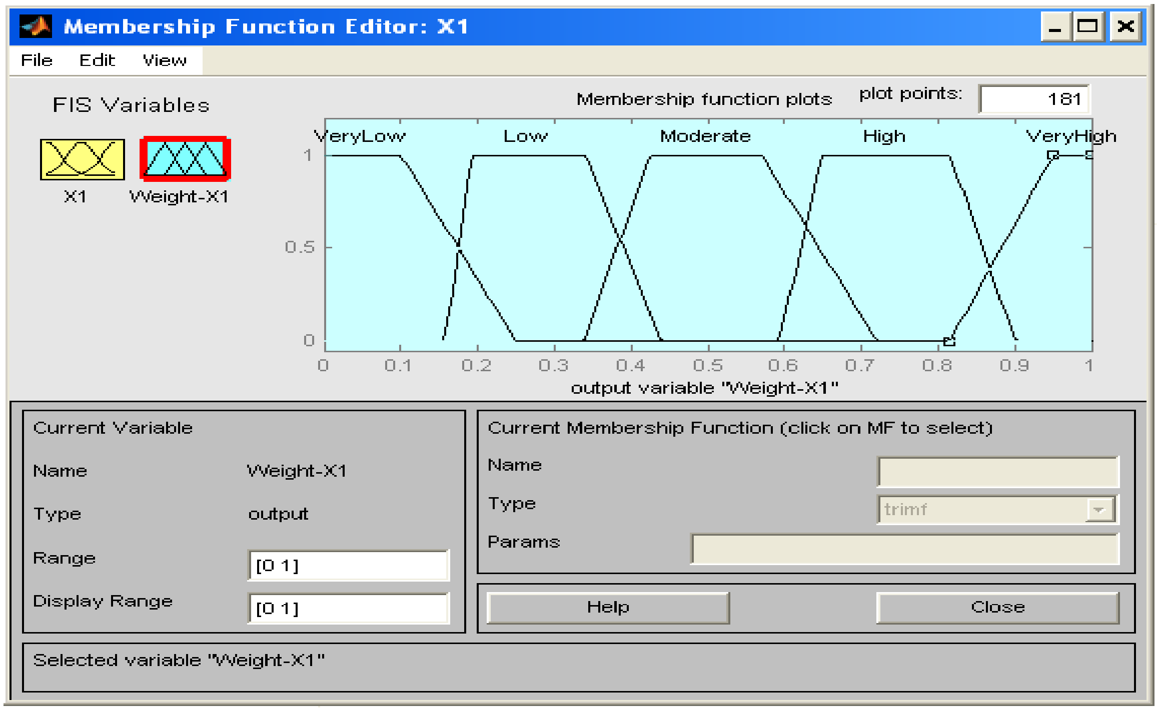
Current Variable
Name: X1
Type: input
Range: [0 100]
Display Range: [0 100]

Type: trimf

Ready

Figure 43: Membership Function Editor for X1 Input

While the user will input the percentage they believe represents their confidence on a scale for 0-100 %, the FIS will *fuzzify* the input according to the five memberships along the five linguistic classifications with a degree of membership to the distribution selected to get an output. This will be aided by the output membership functions also defined by the designer. This can be seen in Figure 44.



plot points: 181

Name Weight-X1

Type output

Range [0 1]

Display Range [0 1]

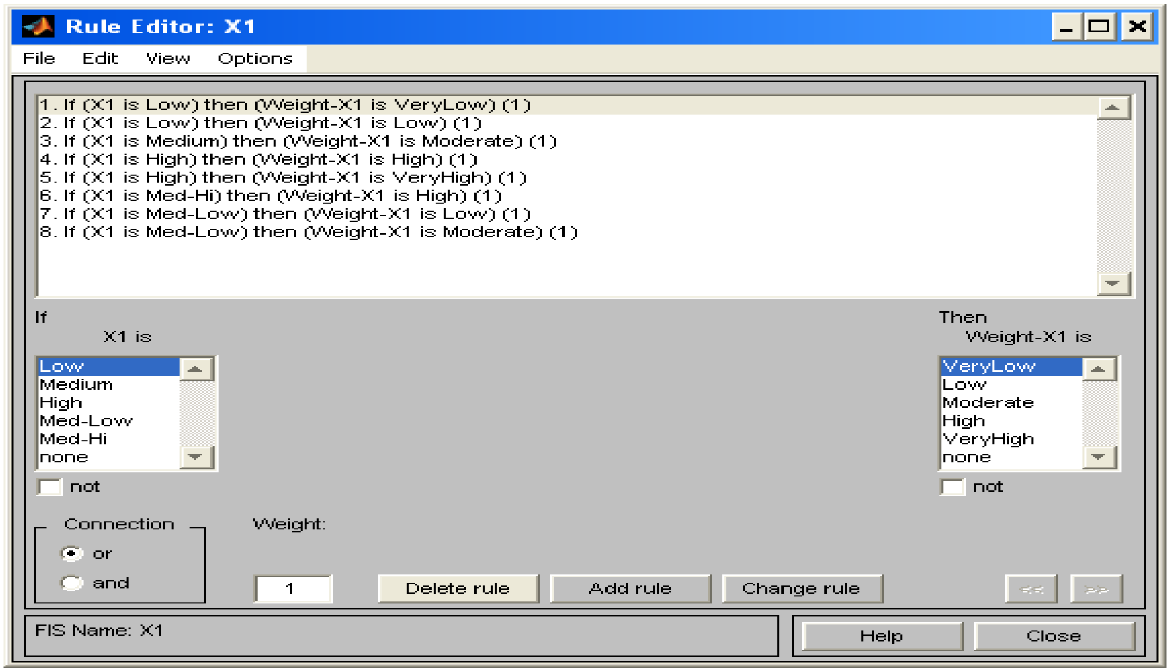
Type trimf

Selected variable "Weight-X1"

Figure 44: Membership Function Editor for X1 Input

As can be seen the output membership functions are represented by trapezoidal shapes. The author felt this output format best represents the range scale of the linguistic classifiers the class of military logisticians would agree with. For example a confidence description of *Very High* likely means that a person is at least 90 % confident and may go as low as around 80 % confidence, depending on who is answering.

This is important because ultimately to determine the quantitative output that most likely resembles the military decision maker thought process, rules must be applied to the input that attempt to reflect this thought. This is accomplished by defining a rule set that is in the form of *if* - *then* statements that utilize primarily the logical operators of *and, or,* and *not*. Creation of these rules is possible when the user defines a consequent and antecedent via the rule editor GUI shown in Figure 45.



1. If (X1 is Low) then (Weight-X1 is VeryLow) (1)
2. If (X1 is Low) then (Weight-X1 is Low) (1)
3. If (X1 is Medium) then (Weight-X1 is Moderate) (1)
4. If (X1 is High) then (Weight-X1 is High) (1)
5. If (X1 is High) then (Weight-X1 is VeryHigh) (1)
6. If (X1 is Med-Hi) then (Weight-X1 is High) (1)
7. If (X1 is Med-Low) then (Weight-X1 is Low) (1)
8. If (X1 is Med-Low) then (Weight-X1 is Moderate) (1)

X1 is: Low

Weight-X1 is: VeryLow

● or

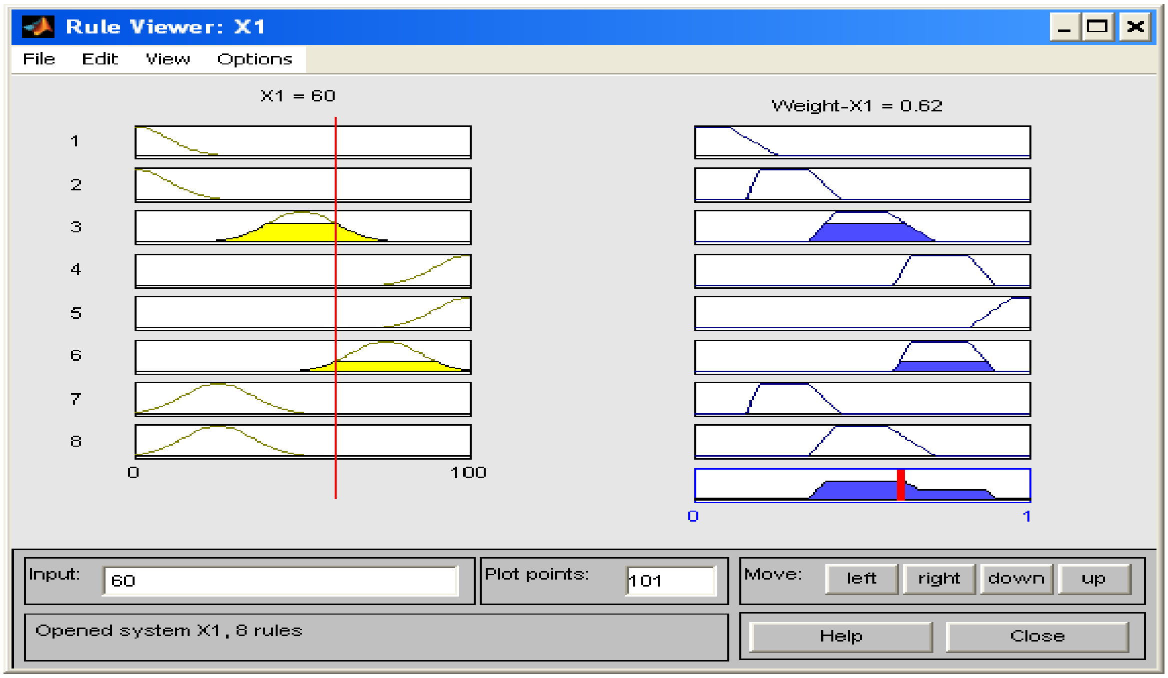
Weight: 1

FIS Name: X1

Figure 45: Rule Editor for X1

Here, eight rules are defined that represent to what membership one of the five linguistic classifications might mean in terms of a persons description of how confident they are in the variable X1's quantitative value. These rules are then utilized to determine the FIS output which should theoretically represent what is meant by a user when they describe linguistically that their confidence in the variable value is *Low, Med-Low, Medium, Med-High,* or *High*. To recap, the logistician would express some level of confidence in *X1*. This value would be *fuzzified* via the membership function defined in Figure 43, the rules specified by the rule editor shown in Figure 45 would be applied, and the output membership functions in Figure 44 would be used to assign a representative output.

A results roadmap of the whole FIS process can be viewed through the Rule Viewer GUI and is displayed in Figure 46. This GUI allows the user to visually see the impact of the membership functions, rules, and the ensuing outcome. While this aids the designer in the FIS generation process this is not the only way the created FIS can be used. Saving the FIS as a *.fis* file allows the creator to execute the FIS by first defining the FIS, i.e. *X1* = *readfis('X1')* and then evaluating the FIS with the command *evalfis(C1,X1)* where C1 is the confidence entered by the user for *X1* and "*X1*" calls *X1.fis*. Execution of the FIS based on a user input of 60 % for their confidence, results in a value of 0.62. This result is shown in Figure 46 and is the overall confidence value that should be applied to the variable X1. Since there are 14 variables, the sum of these weights must be calculated and 0.62 will be normalized for later application with a variant of the OEC formula for the case comparison and selection process by the user in Step 9.



Input: 60

Plot points: 101

Opened system X1, 8 rules

Figure 46: Rule Viewer for X1

The OEC formula was discussed in Section 2.4.3 and the equation is repeated here for reference: $OEC_{Alt,x} = \beta_1 \cdot \frac{Obj_{1,Alt.x}}{Obj_{1,Baseline}} + \beta_2 \cdot \frac{Obj_{2,Alt.x}}{Obj_{2,Baseline}} + ...\beta_N \cdot \frac{Obj_{N,Alt.x}}{Obj_{N,Baseline}}$. As was discussed, OEC is used when a desire exists to minimize or maximize a specific variable that represents the overall goodness of a design. If one were trying to reduce emissions in an aircraft one might look at increasing efficiencies in an aircraft's engine with technology and then determining that ratio multiplied by some weight, $\beta_{1..N}$. If one was trying to maximize the number of enemy killed for the DO platoon scenario, a comparison between the weapons a soldier has today and how the effects of a weapon with increased range would affect the outcome via the OEC formula. Alternatively, the OEC can be used as a way to assess the closeness of variables to one another that result from the calculation of the *kNN* cases to the user's query. Within the OEC are weights that need to be defined. These weights will be created by the FIS to place emphasis on strengths in knowledge the logistician feels they have. Comparing the ratio of the input variables from the query by the user to that of the case retrieved from the database and multiplied by these weights will result in a fitness measure for each of these cases. Ideally, one would prefer a query case to exactly match up to a case in the database, this would result in a ratio of one, but this is a highly unlikely outcome.

In this work the Euclidean distance is used to select the cases, but there is contention by some researchers that this technique does not necessarily select the "best" cases from the database. The reason for this is that the strict minimum difference between the query vector and case vector are used to calculate this metric. Theoretically one could have significantly different cases presented that bared almost no resemblance to the query along several factors, but was still selected. To overcome this potential outcome the variant of the OEC just described will be applied to the selected cases in this step of the ATLAS method to compare these cases on a different scale and determine some level of fitness. Ultimately, the OEC value of one would represent a case that matches the query results exactly and incorporates the confidence expressed by the user. By nature of OEC equation, any ratio between the numerator and denominator will be multiplied by a set of weights that are normalized so that the ideal result would be one. This approach aims to not only incorporate the confidence the user has in the information being entered into the decision support tool, but it also to provide the user with another venue to assess case applicability and provide further insight on factors that potentially effect the logistical need determination and subsequent architecture selection process.

5.3.7 Step 7: Create a Surrogate Model

While utilizing MANA to run simulation cases is beneficial, it is time consuming. Quick design exploration of the operational space as defined by the variable ranges can be accomplished through the use of surrogate models. This topic was discussed in Section 2.8 and 2.11. While there are other options for creating mathematical models of input output data, neural networks will be employed for this step. The characterization of the design space of military operations logistics under the varying needs from humanitarian aid to conventional combat can be described as:

- Vague – Varying operational characteristics based on the situation at hand. For example two different areas of the world are in need of humanitarian aid. Variables that must be considered are level of destruction, existing infrastructure, socioeconomic conditions, ability for indigenous people to assist in the recovery efforts, etc.

- Highly Nonlinear – Very difficult to see any apparent relationship between input variables and outputs. Actions and events can have very disproportionate effects on the outcome.
- Uncertain – While a broad goal is usually apparent, there exists a lot of options and ways to go about accomplishing a mission.
- Complex – There exist many moving factors and pieces that are part of the design space or environment being examined. Modeling these can be challenging, let alone seeing the impact or influence each variable might have, especially because of the nonlinear nature.

Fuzzy Systems and Artificial Neural Networks have been incorporated with great success where human in the loop effects are a part of the environment. Fortunately generating ANN also has been incorporated into a variety of software applications to include MatLab and JMP from SAS.

The author chose to explore NN generation of surrogate models in two ways. The first was a custom tool called Basic Regression Analysis for Integrated Neural Networks (BRAINN) created by Carl Johnson, a fellow graduate student at The Georgia Institute of Technology. This tool allows a user to easily import input and output data into a GUI for creation of neural net formulas. The formulas created for each response are mathematical in nature and meant to emulate the M&S environment. By taking the database created in Excel from the DOE and the output variables of *average Percent Enemy Killed* and *average Helicopter Resupply Flights* over the 5 iterations of each DOE run, the author was able to execute BRAINN with a variety options to include 13 training algorithms and several training options. BRAINN allows the formulas generated to be saved in a variety of usable formats for post processing in software like JMP, Excel, or MatLab; the *averaged Standard Deviation* from the *Percent Enemy Killed* MOE was also included since it was determined there was no added expense in time to the calculation process for this additional formula.

As was discussed earlier and in Appendix D, neural network creation are often a balance between art and science. Different training algorithms result in a wide range of results, as well as the number of hidden nodes that a NN incorporates to arrive at a suitable surrogate. BRAINN allows the user to explore these options and saves the best results based on sum square error between the training and the test/validation data. While the use of NN can be categorized as a "black box" process, it has been accepted in the research community as an acceptable downfall because neural networks do very well in representing the highly nonlinear and complex design spaces being investigated in today's research community.

The author executed BRAINN with two training algorithms, the Levenberg-Marquart with Bayesian Regulation and the Gradient Descent with Momentum Adaptive Learning Rate. These two algorithms resulted in the best two fits, i.e. highest R^2 when analyzing the NN outputs from BRAINN. For the NN creation process, 10% of the data was excluded to be used for validation purposes on the resulting NN surrogate. This is represented by the blue dots in Figure 47, Figure 48, and Figure 49. The Levenburg Marquart algorithm performed slightly better than the Gradient Descent algorithm for both the *average Percent Enemy Killed* and *average Helicopter Resupply Missions* while the Gradient Descent performed better for the *average Standard Deviation* associated with the mean *Percent Enemy Killed*. The output results from BRAINN for *Percent Enemy Killed* are displayed in Figure 47. The R^2 value or the proportion of the variability in the data set that is accounted for by a statistical model for this response is 90.8 %. Generally one would like to see a surrogate model R^2 value that is in the high 90 percentile if not greater than 99 %. This would mean that the formula generated by the NN extrapolation models the environment with a high degree of accuracy. Unfortunately this environment as previously described is highly nonlinear and encompasses variability due to human interactions and intangibles that are extremely difficult to capture from the

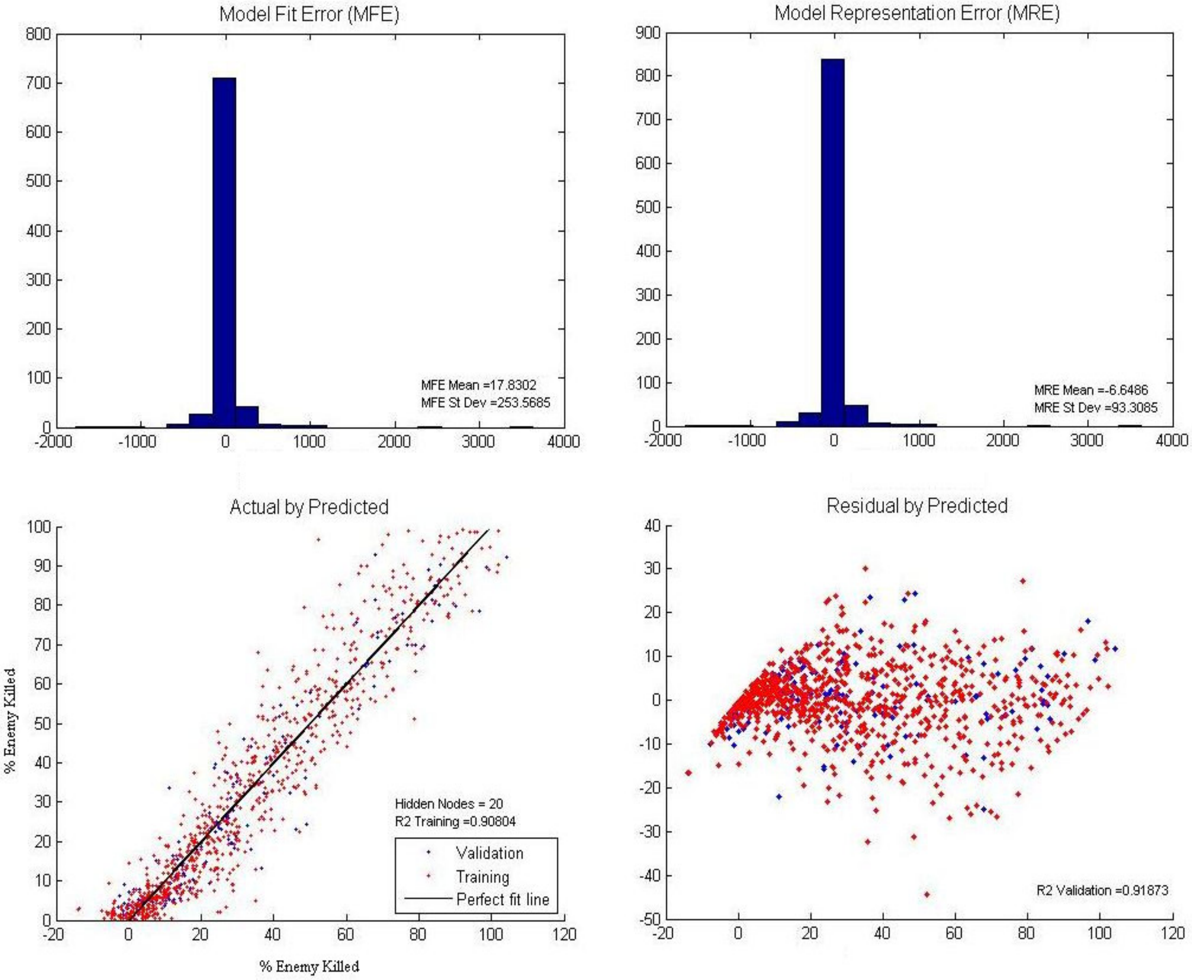


Figure 47: BRAINN NN Output Results for *% Enemy Killed*

MANA ABM. The author chose to accept this fact because although a very accurate model of the DO platoon's operational environment would be nice, no real world environment that involves the intangibles of human emotions, ingenuity, resolve etc. will ever be likely or usable because of these characteristics. Therefore an R^2 of 90 % for the surrogate should be an acceptable value for design space exploration by military logisticians or DM's. Figure 48 shows the results for the NN surrogate applied to the MOE of *Helicopter Resupply Missions* required. The R^2 value for this surrogate model is

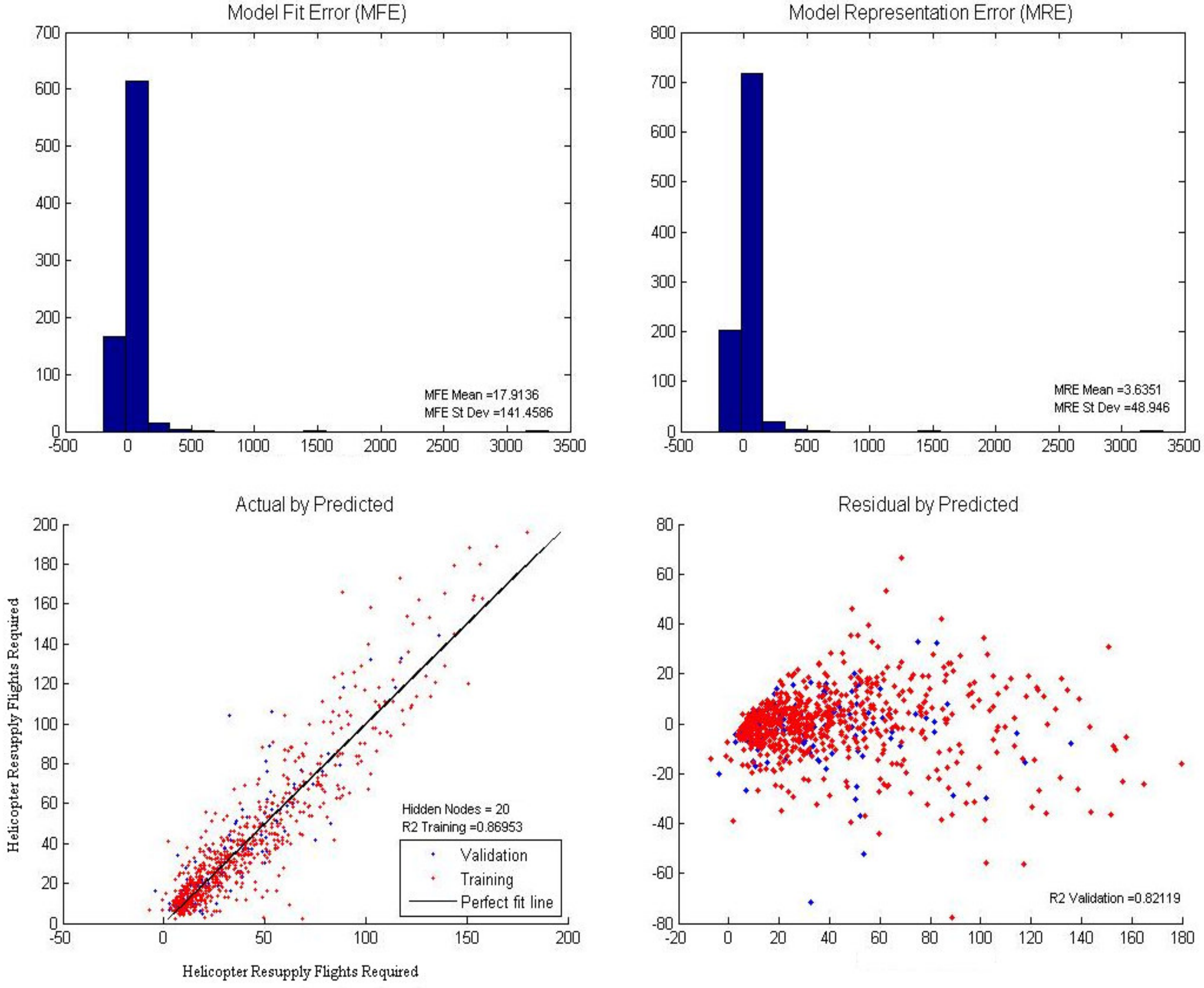


Figure 48: BRAINN NN Output Results for Helicopter Flights Required

0.8695, not even in the ninety percentile range, but again based on the characteristic of the ABM agents and opportunity for a variety of different outcomes based on the five iterations, the author felt this value was acceptable for what was trying to be modeled. The output for the standard deviation was extremely bad for the NN surrogate and although not usable to represent the information, the information was not a direct result of the model and was again included because the cost of including it was not detrimental to running BRAINN to see the result. The resulting output can be seen in Figure 49.

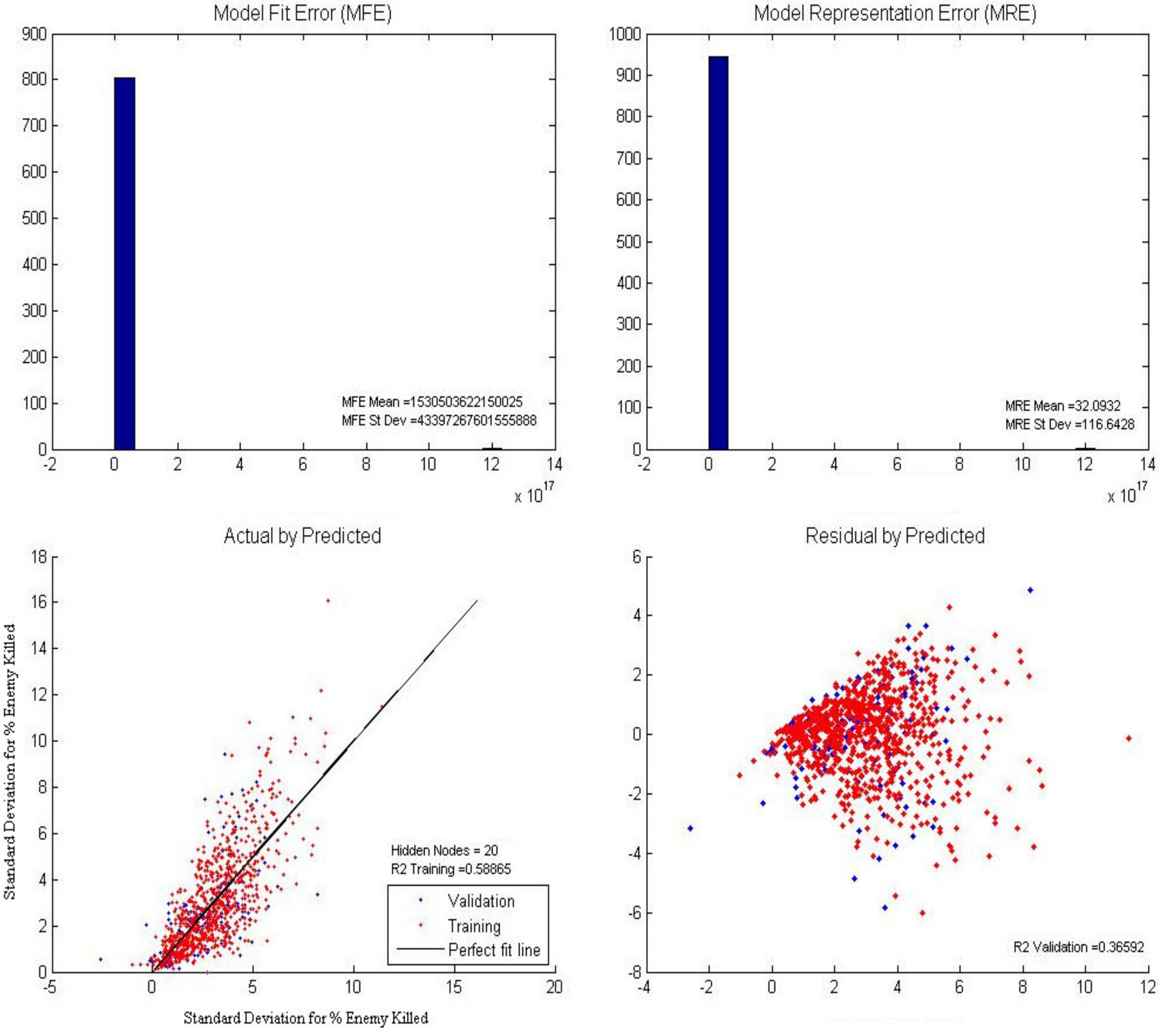
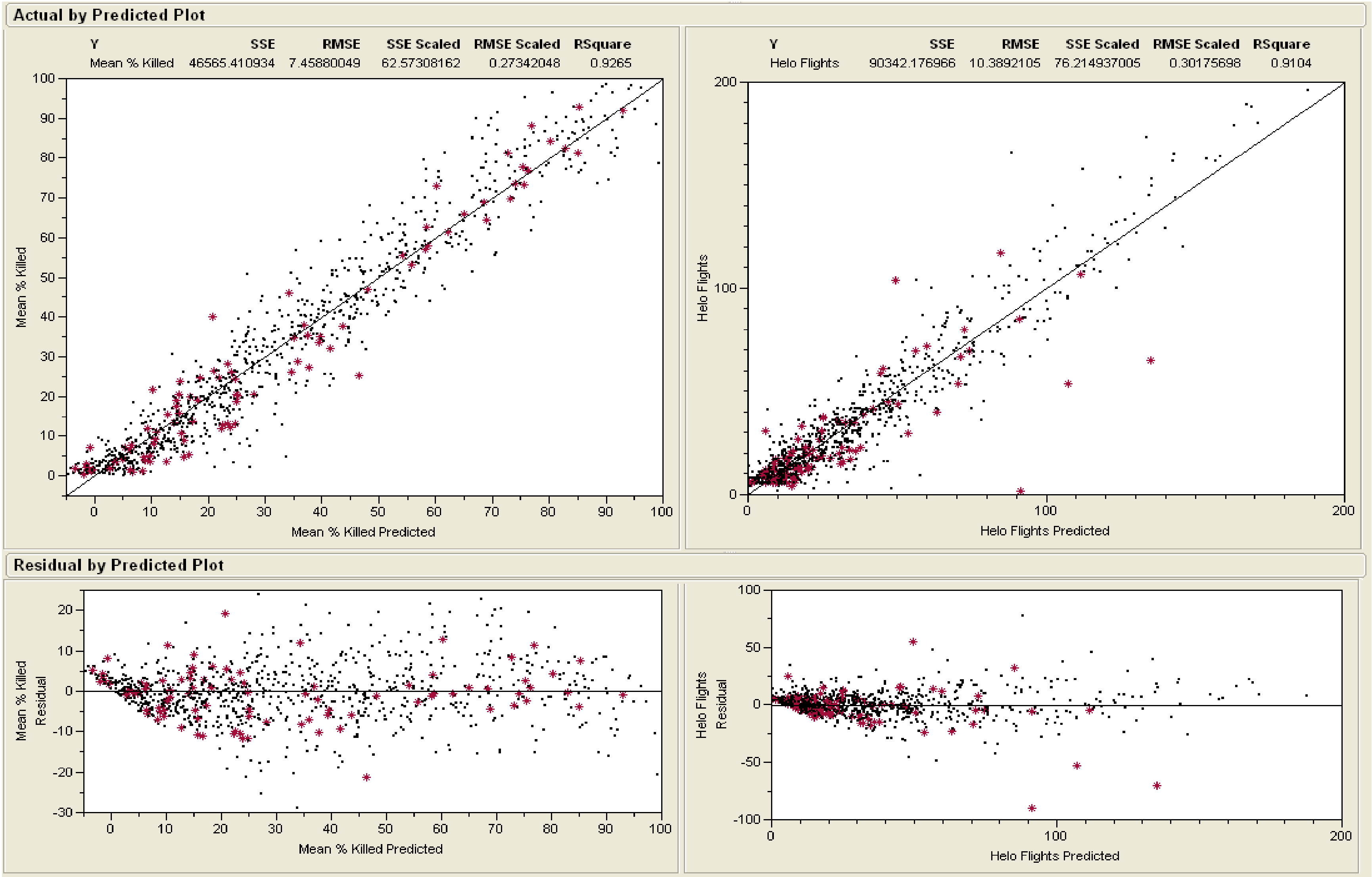


Figure 49: BRAINN NN Output Results for Standard Deviation of % Enemy Killed

The author also explored analyzing the data to create a surrogate the statistical software package JMP, which also has neural net capabilities. The results were surprisingly better than those produced by BRAINN. Executing JMP's NN modeling function under the *Analyze* tab enables the user to specify the number of hidden nodes to explore singularly or the user can use the drop down box from the red NN menu arrow and select *Sequence of Fits*. This has JMP conduct a series of NN extrapolations based on user defined criteria. The author performed this in similar fashion to that which was performed in BRAINN. Hidden nodes were explored by starting at five and incremented by one, going up to 15 nodes. The best result was three hidden nodes with a penalty function imposed of 0.001 and an overall R^2 equal to 91.846 % across the two MOEs; the first MOE had an R^2 of 92.65% and the second MOE had an R^2 of 91.04 %. This reinforces the comment that neural network generation often times is more of an art than a science. A look at the underlying equations generated by JMP and BRAINN, which is MatLab based, showed an extremely more complex equation generated by JMP and is likely the cause for the better R^2 value for the model fit. Neural networks often have a problem of over fitting the data used to create the surrogate, such that the surrogate is only capable of accurately being used on those data points. To check the fit of the JMP NN surrogate, 10 % of the data was also excluded, as in BRAINN, for validation purposes. The results of the JMP analysis for the outputs *Percent Enemy Killed* and *Helicopter Resupply Missions* are show in Figure 50. The black dots are the data points used to create the surrogate and the red asterisks are the validation points. An examination of the resulting plots shows a relatively good fit for the data extracted from the MANA database and the user is confident that the surrogate should be useful for



Y	SSE	RMSE	SSE Scaled	RMSE Scaled	RSquare
Mean % Killed	46565.410934	7.45880049	62.57308162	0.27342048	0.9265

Y	SSE	RMSE	SSE Scaled	RMSE Scaled	RSquare
Helo Flights	90342.176966	10.3892105	76.214937005	0.30175698	0.9104

Figure 50: JMP NN Results for % Enemy Killed and Resupply Helicopter Flights

application to the ATLAS method in Steps 8 and 9. The author chose to utilize the NN formulas generated by JMP for the surrogate MANA model because of their slightly better fits as shown in Figure 50. A representation of the neural net equation with its three hidden nodes from JMP is displayed in Figure 51.

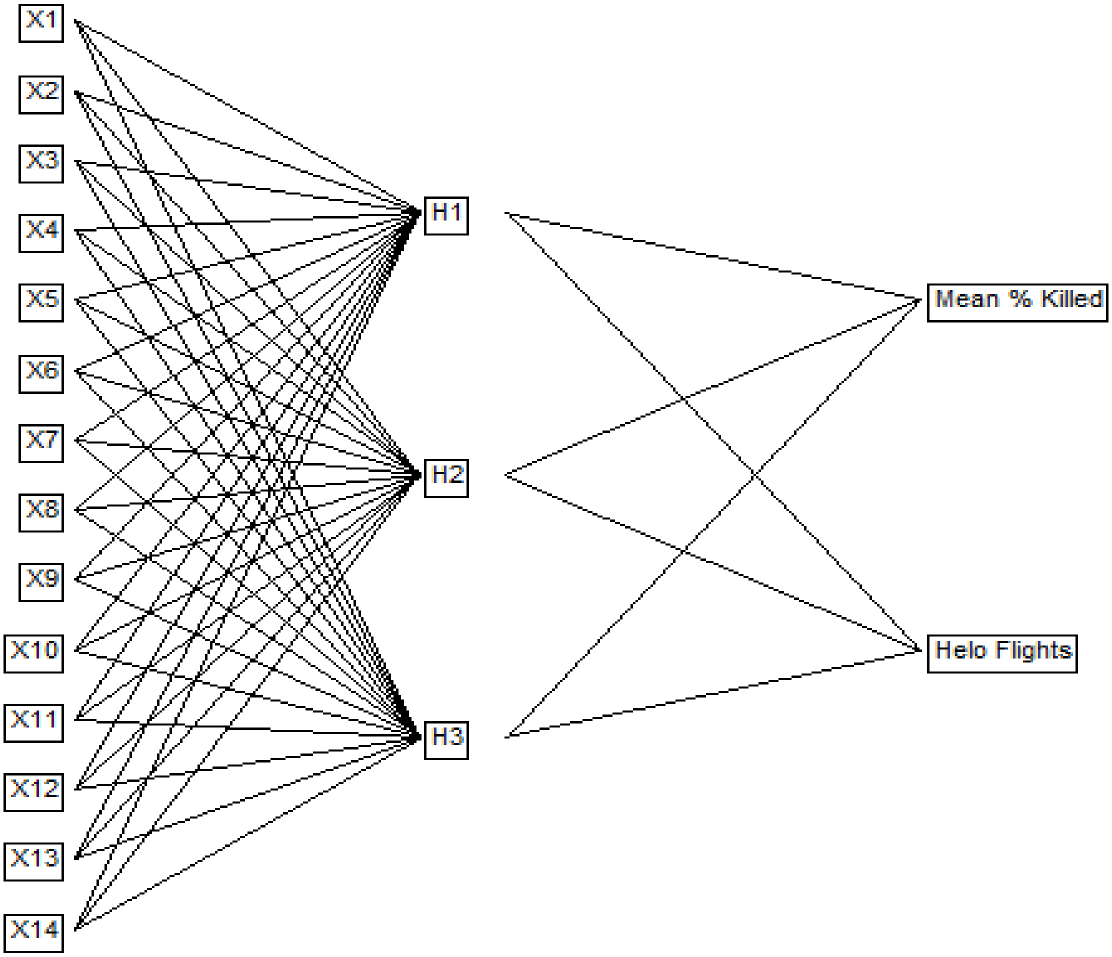


Figure 51: DO Platoon Neural Net Representation from JMP

5.3.8 Step 8: Visualize and Explore the Design Space

The author selected two classes of users that could benefit from the ATLAS method and the visualization techniques selected to represent the design space representative of the DO platoon being modeled. They are the logistician in the field who needs a quick viable answer so that he or she can quickly decide the logistics architecture best suited for the operational environment. The other user is the high level logistician who might want to see down to the details in terms of low level operational needs, but also needs be able to aggregate up for a holistic view to determination overall logistics architecture requirements. To accommodate these two classes, the author pursued design space exploration via JMP for the high level logistician and then created a MatLab GUI for the operational logistician that interfaces with the Excel database to quickly draw on information and display useful data to the logistician or DM.

5.3.8.1 JMP

The statistical software package known as JMP allows a designer to explore data for the design space in a variety of ways. Among them is the capability to visualize the impact of the input variables on the outputs or responses via tools such as the prediction profiler, contour profiler, surface profiler, Pareto plots, etc. Each one of these visual aids in their own way presents the designer with a graphical orientation of the input factors in relation to the responses, in this case the two MOEs. Using these tools allows the designer to see the sensitivities that the input factors, whether large or small, have on the responses.

The prediction profiler provides an overall visualization of the space depicted by the NN equations and is shown in Figure 52. With this tool the designer can start to see trends and impacts certain input factors have on the response. Often times the profiler provides intuitive interaction behaviors between the inputs and outputs, but it can also show behaviors not expected or can display the coupling of input factor effects not seen by general observation. The user can adjust the inputs and see the impacts to the other input factors as well as the responses in a dynamic fashion. This capability is referred to

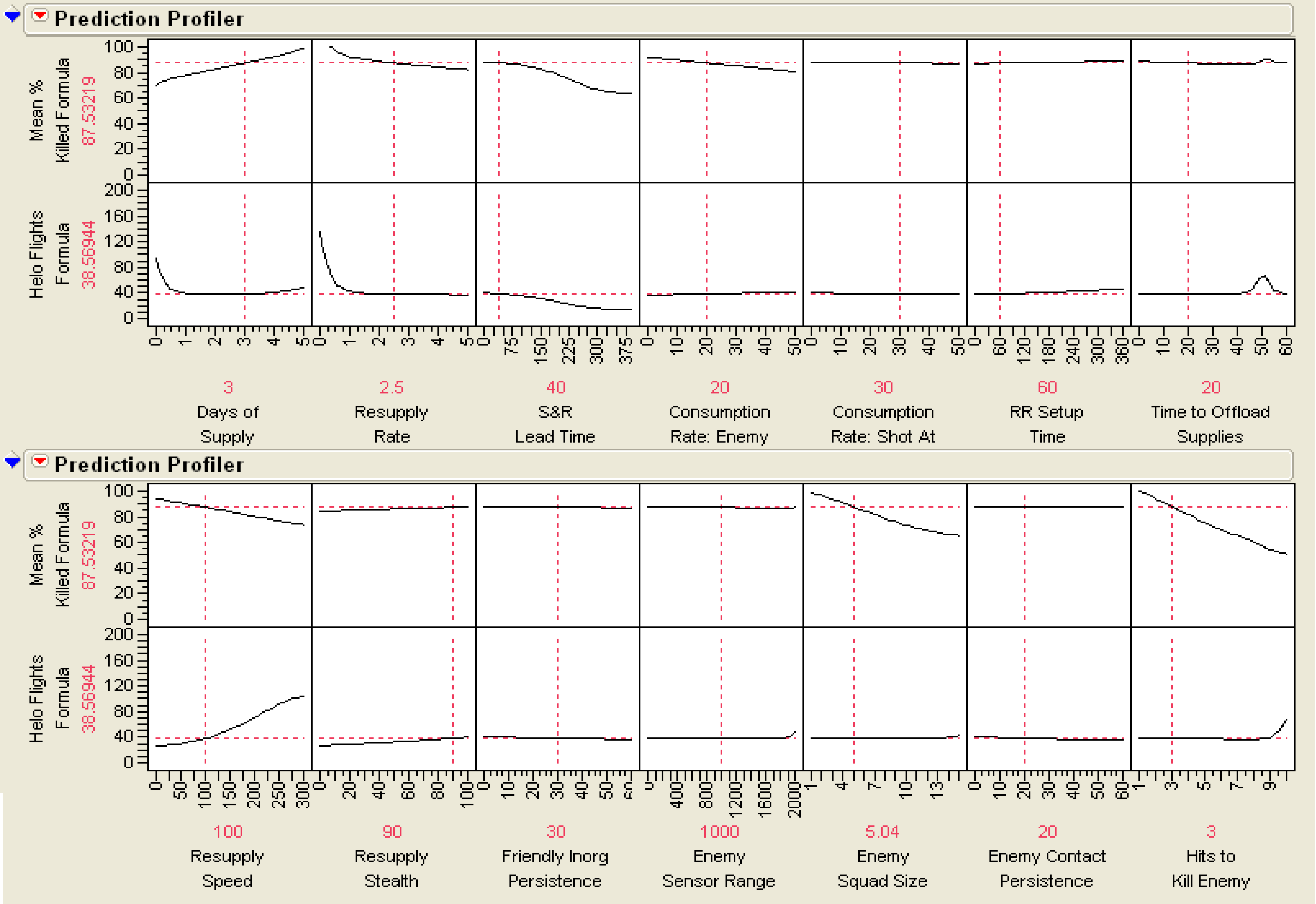


Figure 52: DO Platoon Prediction Profiler

as *what-if* exploration and enables the designer to virtually explore options or COA's by seeing how sensitive the responses are to changes in the input factors and the predicted result of those changes as reflected in the responses. Using this tool can effectively provide a DM with increased confidence in decisions they make that are backed by quantifiable data and not just intuition; thus providing justification as to why a particular COA was selected over another. The profiler in Figure 52 shows that several relationships exist, positive and negative as indicated by the slope of the prediction lines, for variables X1, X2, X3, X4, X8, X12, and X14 in relation to the two MOE's. Variables X5, X6, X10, and X13 have relatively flat lines and have relatively no effect at the current settings indicated by the red dashed vertical sensitivity lines; X7 and X9 have some impact but are negligible. An example of a *what-if* exploration would be to keep variable X8, *Resupply Speed*, constant at 100 m/s, equivalent to about 194 knots. The top speed of the MV-22, pictured in Figure 53, is 240 knots. It is the USMC's main mode

Figure 53: MV-22 Osprey: USMC Transport Aircraft

for air transportation and logistics resupply, so this speed is well within the range of this aircraft's capabilities and anything under it should be feasible. Now the DM can slide the sensitivity sliders (red dashed lines) along the range axes for each variable to try and determine the best conditions that will promote success in the MOEs with a high *Percent Enemy Killed* and minimal *Helicopter Resupply Flights*. Alternatively, a designer could hold constant all known factors and then only explore those variables that are ambiguous to see the likely worst or best case scenarios facing the DO platoon for this mission.

Another useful tool for the user is the Pareto plot. For instance referring to Table 16, a screening test is performed on the responses and the result is Figure 54 which shows

Table 16: Input Variables for MANA Model

X1	DO Squads Days of Supply	X8	Resupply Speed
X2	Scheduled Resupply Rate	X9	Resupply Stealth
X3	Sense and Respond Lead Time	X10	Friendly Inorg. Sensor Persistence
X4	Consumption Rate: Enemy Contact	X11	Enemy Sensor Range
X5	Consumption Rate: Shot at	X12	Enemy Squad Size
X6	Rapid Request Setup Time	X13	Contact Persistence Enemy
X7	Time to conduct Resupply Mission	X14	Enemy Hits to Kill

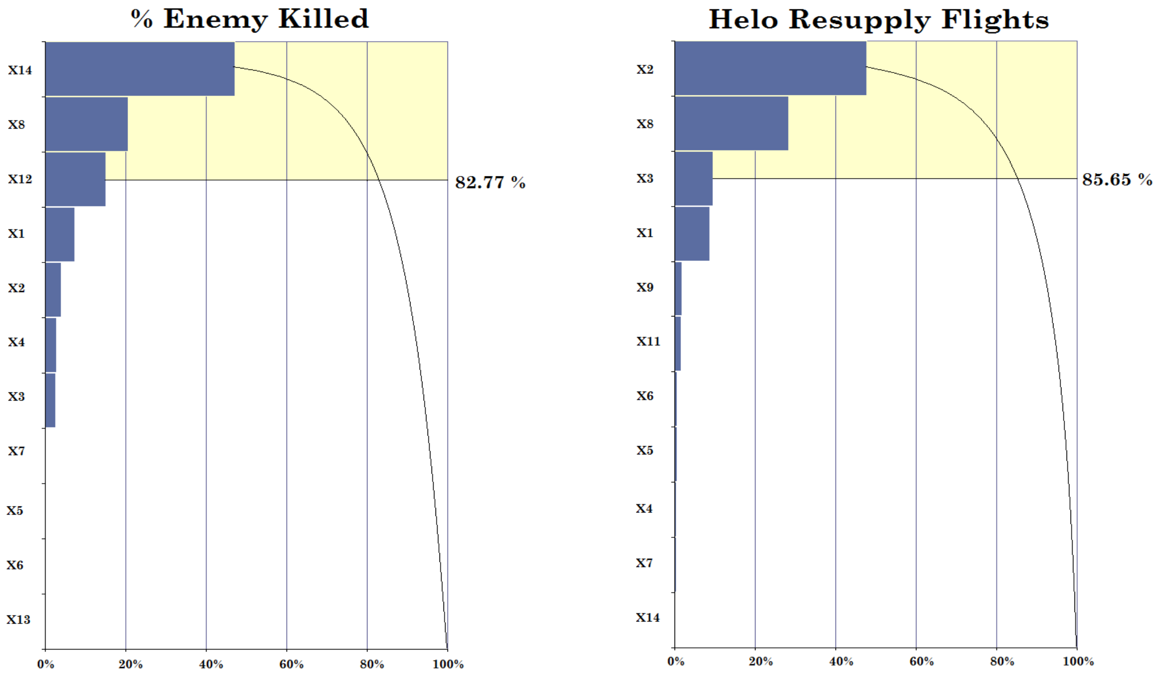
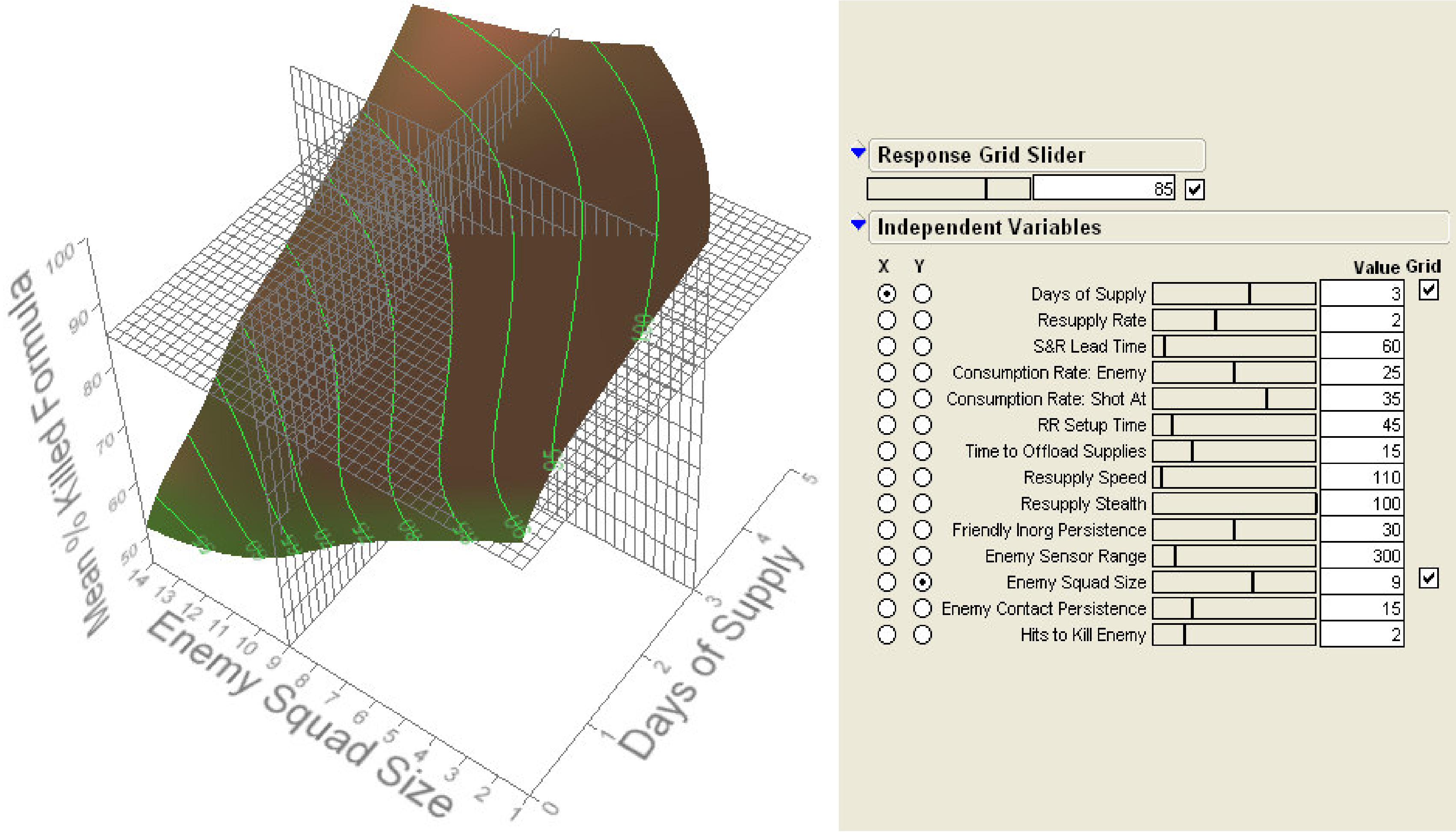


Figure 54: Pareto Plot showing the relative Influence of Input Factors on Responses

that 80% of the variability of the MOE's comes from primarily three factors for each response. This plot tells the viewer which input variables have the most effect on the responses *Percent Enemy Killed* and *Helicopter Resupply Flights*. Using this plot provides the observer an idea of which variables need to be looked at first to see what improvements or efficiencies can be realized.

Referring to Figure 54, the only one factor each response has in common is *X8*, with *X14* and *X12* contributing the remainder of the 80 % sum of the effects to *Percent Enemy Killed* and *X2* and *X3* contributing towards the number of *Helicopter Resupply Flights* needed for the mission. This data can now be used by a logistician to focus on these particular variables for future analysis because they are the bulk contributors to these particular MOEs. Up until this point trying to select which variable had the most importance was almost impossible. For the first MOE, *X12* and *X14* are defined as noise factors that fall under the enemy characteristics. While out of the logisticians control, this information tells the logistician that getting as much intelligence in these two areas is critical to the DO platoon's success of eliminating as many enemy combatants as possible. For the second response, the DM can look at the scheduled resupply rate and sense and respond lead times which theoretically are in the DM's control. Use of the Pareto plots does no indicate a DM should exclude the factors below the 80% line, as these variables of importance will change as more data is added or as the operational scenario changes, but it does provide a focus where there was likely none. Looking below this line also tells the DM what the next contributing factors are and where further emphasis might be important during the operation.

JMP also allows a user to look at detailed interactions between two factors alone with respect to the response. One such tool is the surface profiler shown in Figure 55. This example shows a comparison between *Squad Days of Supply (X1)* in days and *Sense* and *Enemy Squad Size (X12)* in minutes. As *X1* increases it means the squad has more supplies on hand and can eliminate more of the enemy because they have the supplies and flexibility to react. As the amount of supplies on hand decreases, the platoon has less and this restricts their success in this area. When X12 is introduced we see a decrease in the platoon's ability to kill the enemy as the squad size increases. This makes intuitive sense to the observer and gives some sense of validation that the surrogate model is capturing the effects of the input factors on the response of *Percent Enemy Killed*. It is important to note that the surface can and will dynamically change based on the variable and its settings located on the right of the plot. JMP also enables the user to further focus on specific cases of interest when the user initiates the grid functionality. Referring to Figure 55, the user desires at least 85 % of the enemy killed. The cases that represent that desire create the contour above this grid along the Z axis of the plot. Based on an initial three days of supply on hand, the observer can see that enemy squad size can not exceed nine combatants if this goal is to be met. While helpful, this is only true if all other input factors remain unchanged. So if the DM is fairly confident about all of the other variables and thinks they will remain relatively fixed, the DM can see the impacts of these two variables on the response and what settings should be maintained to achieve a prescribed success rate. This setting also shows the designer that if intelligence is saying enemy squad size is grater than nine, more that three days of supplies need to be provided to the DO platoon to maintain the at least 85 % enemy killed metric.



Response Grid Slider

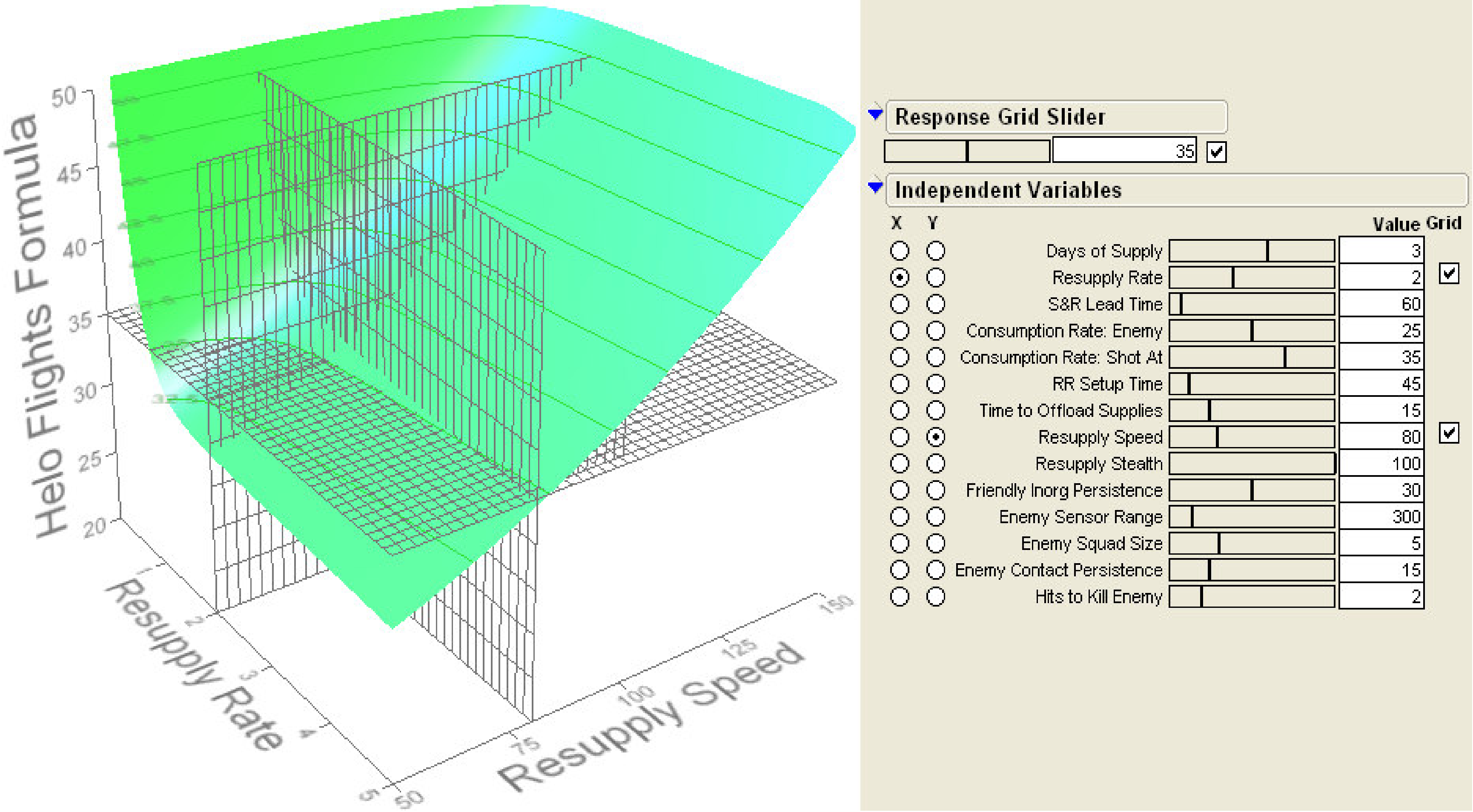
85 ☑

Independent Variables

X	Y		Value	Grid
●	○	Days of Supply	3	☑
○	○	Resupply Rate	2	
○	○	S&R Lead Time	60	
○	○	Consumption Rate: Enemy	25	
○	○	Consumption Rate: Shot At	35	
○	○	RR Setup Time	45	
○	○	Time to Offload Supplies	15	
○	○	Resupply Speed	110	
○	○	Resupply Stealth	100	
○	○	Friendly Inorg Persistence	30	
○	○	Enemy Sensor Range	300	
○	●	Enemy Squad Size	9	☑
○	○	Enemy Contact Persistence	15	
○	○	Hits to Kill Enemy	2	

Figure 55: Variation between *Enemy Squad Size & Days of Supply*

For the response representing the number of *Helicopter Resupply Flights* in Figure 56, the plot shows the DM that the amount of time between scheduled resupply has almost no effect on helicopter flights as long as the resupply schedule remains greater that approximately 1.5 days. If the time is any shorter than that the logistician can expect to see a sharp increase in the number of helicopter flights the aviation assets from the MEU will have to conduct to keep the platoon resupplied. Intuitively increasing speed that the aerial assets fly at should increase the amount of flights the MEU is capable of making over the course of the seven day mission; this is also reflected in this plot. If the logistician was limited in resources and calculated that the MEU could only provide 35 flights as indicated by the mesh grid on the Z axis and the DO platoon's resupply package consisted of two days worth, the resupply asset would have to fly at a minimum of 80 m/s or approximately 161 knots. This means either the MV-22 or the CH53-E could be used since their top speeds are 275 knots and 170 knots respectively. The DM can also see that speed is a critical factor and the helicopters, especially the CH53-E will either be flying close to their top speed or at a rate not conducive to efficient fuel consumption. This potentially means more fuel being expended overall for the seven day mission since the aircraft must travel at less than ideal fuel efficient speeds to conduct resupply activities. Overall, these plots allow the DM to explore and see the impact two factors of interest have on the responses. By using the prediction profiler and the contour plots, trends or interesting behaviors can be identified and then the logistician or DM can explore the impacts of these factors on the specific MOEs for the investigated scenario.



Response Grid Slider

35 ☑

Independent Variables

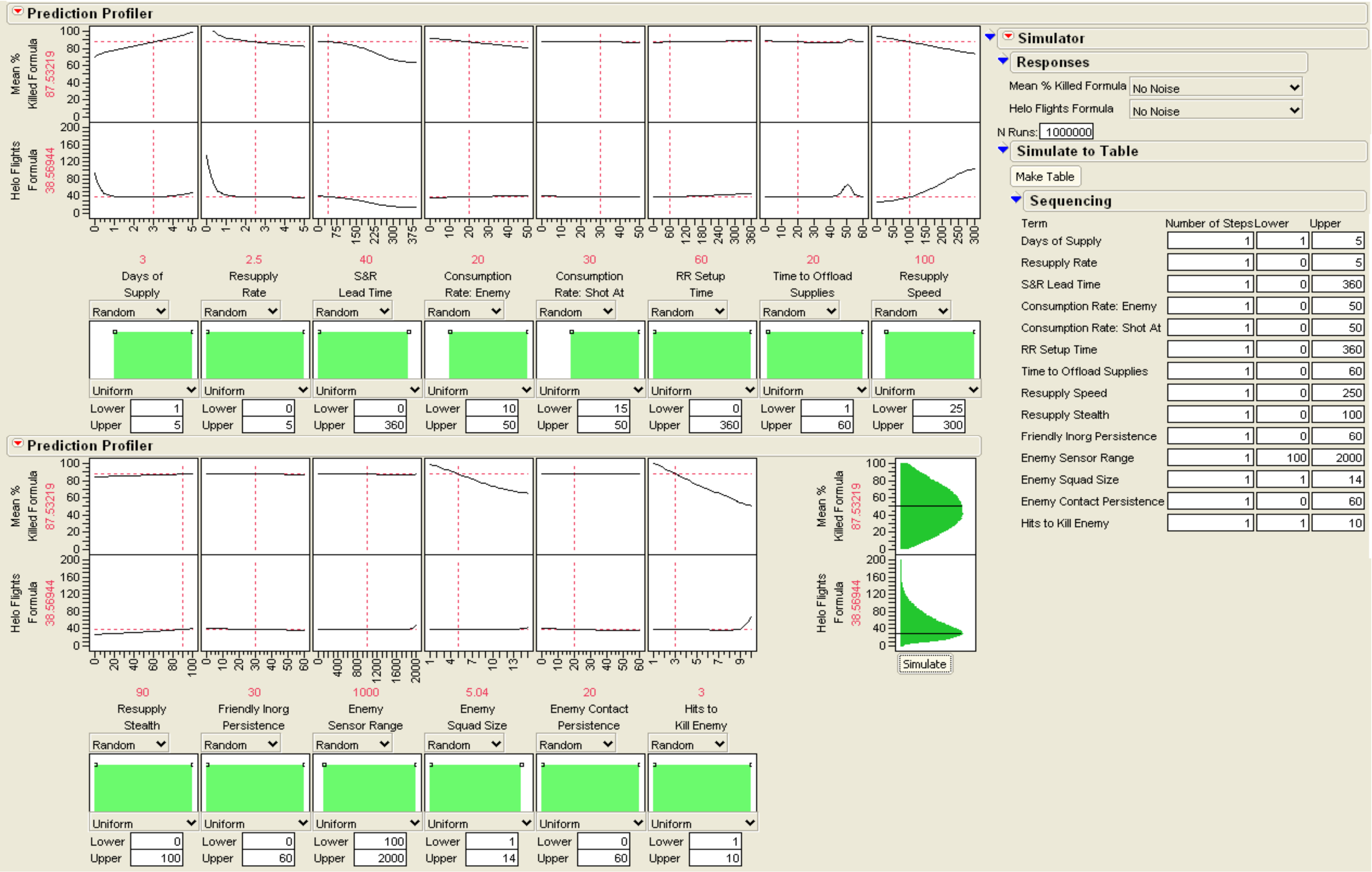
X	Y		Value	Grid
○	○	Days of Supply	3	
●	○	Resupply Rate	2	☑
○	○	S&R Lead Time	60	
○	○	Consumption Rate: Enemy	25	
○	○	Consumption Rate: Shot At	35	
○	○	RR Setup Time	45	
○	○	Time to Offload Supplies	15	
○	●	Resupply Speed	80	☑
○	○	Resupply Stealth	100	
○	○	Friendly Inorg Persistence	30	
○	○	Enemy Sensor Range	300	
○	○	Enemy Squad Size	5	
○	○	Enemy Contact Persistence	15	
○	○	Hits to Kill Enemy	2	

Figure 56: Variation between *Resupply Rate* & *Helicopter Resupply Speed*

]

Relatively new to JMP is the ability to conduct Monte Carlo simulations (Refer to Section 2.9). This tool allows the DM to examine different distributions on the input variables. As knowledge increases over the course of the scenario, the DM can adjust these distributions to quickly run the model to help predict the outcome. Figure 57 shows what a notional initial condition scenario might entail. Running the simulation 1,000,000 times, approximately five minutes, with uniform distributions about the ranges results in a distribution of likely outcomes for both of the responses as seen in Figure 57.

The *Percent Enemy Killed* response has a normal type distribution with the mean approximately at 50% and the number of *Helicopter Resupply Flights* has a skewed normal distribution, with a mean at around 30. The current settings inform the DM that these particular mission settings should result in about 88 % of the enemy eliminated and approximately 39 flights over the course of the seven day mission to keep the DO platoon adequately supplied. What can also be surmised from this tool is that there appears to be a high propensity for at least 30 flights occurring over the course of any mission. If a logistician not knowing anything about the operational scenario specifics wanted to at least prepare for a short notice mission, it would be logical to assume that at least 30 flights will be needed. This also can be used by the high level DM or logistic planner in terms or resource needs and allocation of assets to meet these needs if more than one DO platoon was needed for an operation. These insights should aid in initial planning activities and gives a starting point for forecasting baseline requirements for future missions that will be asked of the MEU's organic aviation and logistics combat elements.



Days of Supply	3	Random	Uniform	Lower 1	Upper 5
Resupply Rate	2.5	Random	Uniform	Lower 0	Upper 5
S&R Lead Time	40	Random	Uniform	Lower 0	Upper 360
Consumption Rate: Enemy	20	Random	Uniform	Lower 10	Upper 50
Consumption Rate: Shot At	30	Random	Uniform	Lower 15	Upper 50
RR Setup Time	60	Random	Uniform	Lower 0	Upper 360
Time to Offload Supplies	20	Random	Uniform	Lower 1	Upper 60
Resupply Speed	100	Random	Uniform	Lower 25	Upper 300
Resupply Stealth	90	Random	Uniform	Lower 0	Upper 100
Friendly Inorg Persistence	30	Random	Uniform	Lower 0	Upper 60
Enemy Sensor Range	1000	Random	Uniform	Lower 100	Upper 2000
Enemy Squad Size	5.04	Random	Uniform	Lower 1	Upper 14
Enemy Contact Persistence	20	Random	Uniform	Lower 0	Upper 60
Hits to Kill Enemy	3	Random	Uniform	Lower 1	Upper 10

Mean % Killed Formula: No Noise

Helo Flights Formula: No Noise

N Runs: 1000000

Term	Number of Steps	Lower	Upper
Days of Supply	1	1	5
Resupply Rate	1	0	5
S&R Lead Time	1	0	360
Consumption Rate: Enemy	1	0	50
Consumption Rate: Shot At	1	0	50
RR Setup Time	1	0	360
Time to Offload Supplies	1	0	60
Resupply Speed	1	0	250
Resupply Stealth	1	0	100
Friendly Inorg Persistence	1	0	60
Enemy Sensor Range	1	100	2000
Enemy Squad Size	1	1	14
Enemy Contact Persistence	1	0	60
Hits to Kill Enemy	1	1	10

Figure 57: Simulate (Monte Carlo) Tool for DO Platoon

Figure 58, below, displays an irregular shape of contours that depicts the density of the cases in the database for the two responses for the DO platoon analysis. There exists a high density of the cases in the lower left hand corner of the plot as indicated by the red colored contours. Unfortunately this also shows that the bulk of the cases result in a low percentage of the enemy killed. As the contours spread from the origin there are less and less cases as indicated by the diminishing number of cases populating the space.

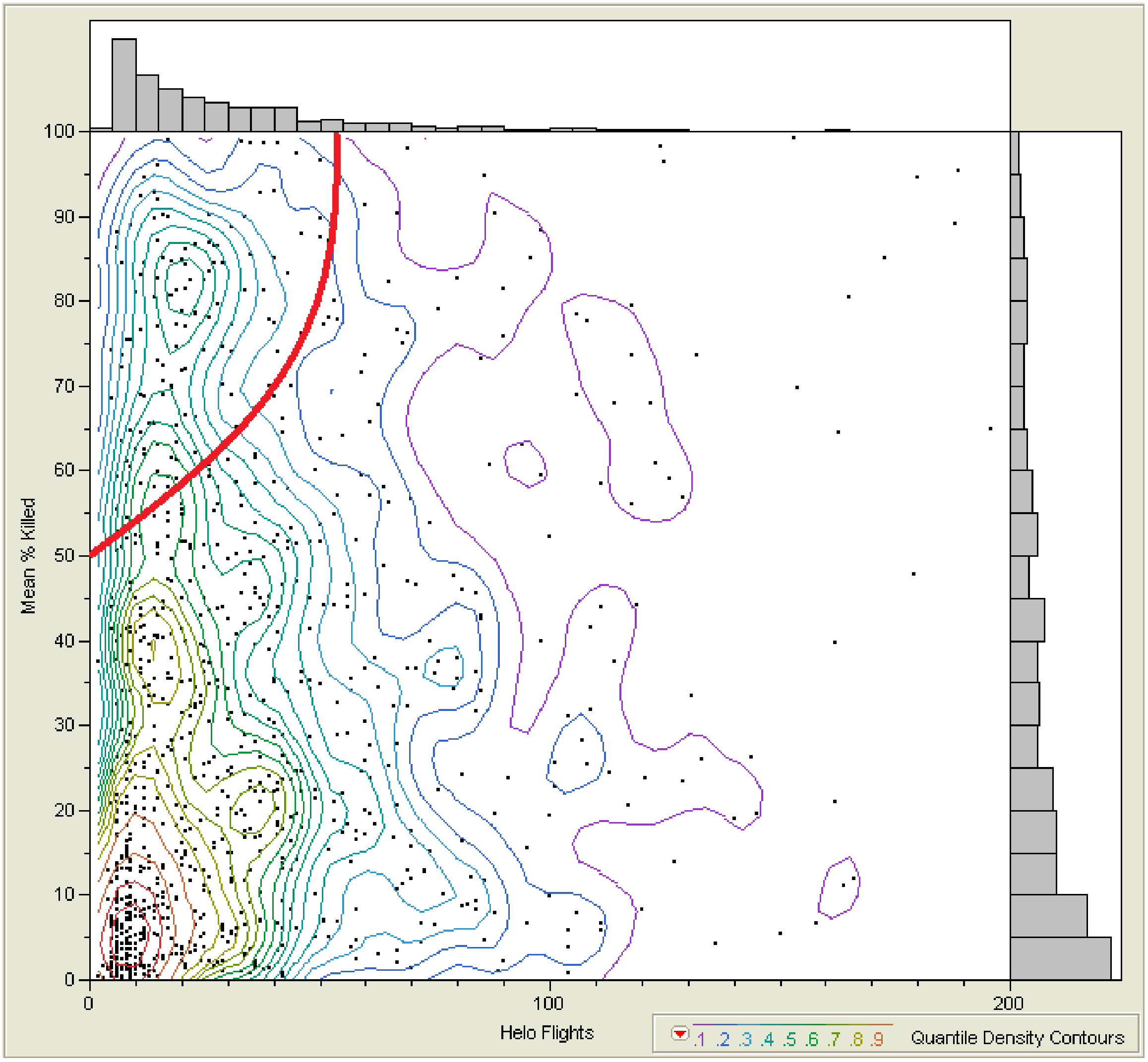


Figure 58: Contour Chart of *Percent Enemy Killed* vs. *Number of Resupply Flights*

Visually the user can see there are cases that potentially fall within a notional and arbitrary band of feasible and acceptable outcomes. This is depicted by the red curved line that begins at the 50 % point on the Y axis and curves up and connects to an imaginary point where approximately 50 resupply helicopters would be employed to support the DO platoon for its missions. These points and the line were uniquely defined by the author. The depiction of this line and the area within it encapsulate the specific cases that fall with in the criteria of at least 50 % of the enemy killed and no more that 50 helicopter resupply flights conducted. The user could then highlight these points and see the specific input variable settings and corresponding responses to further evaluate potential courses of action. The purpose of the curve to the line is to reflect that as a DM one would likely be amicable to conducting more flights as the *Percent Enemy Killed* increases, but would be inclined to not allocate aviation assets as this number decreases.

Figure 59 is a variant of the density plot above that enables the user to place an ellipse of varying percentage to see where each of the cases fall. This plot also shows the correlation between the two MOEs. The pair wise correlation between the two responses is 0.2109 and the Pearson's rho non-parametric correlation is 0.3042. This tells the user that a small correlation between these two responses exists and there is some dependence associated between the two responses. Beyond this, the plot is not unlike the density plot in Figure 58 except this figure shows the histogram and counts associated with the plot. This reinforces the observation that a high density of the cases are congregated about a low percentage of enemy killed and low number of helicopter resupply flights. The tool allows the user to select the density of flights they are most interested in via the ellipses.

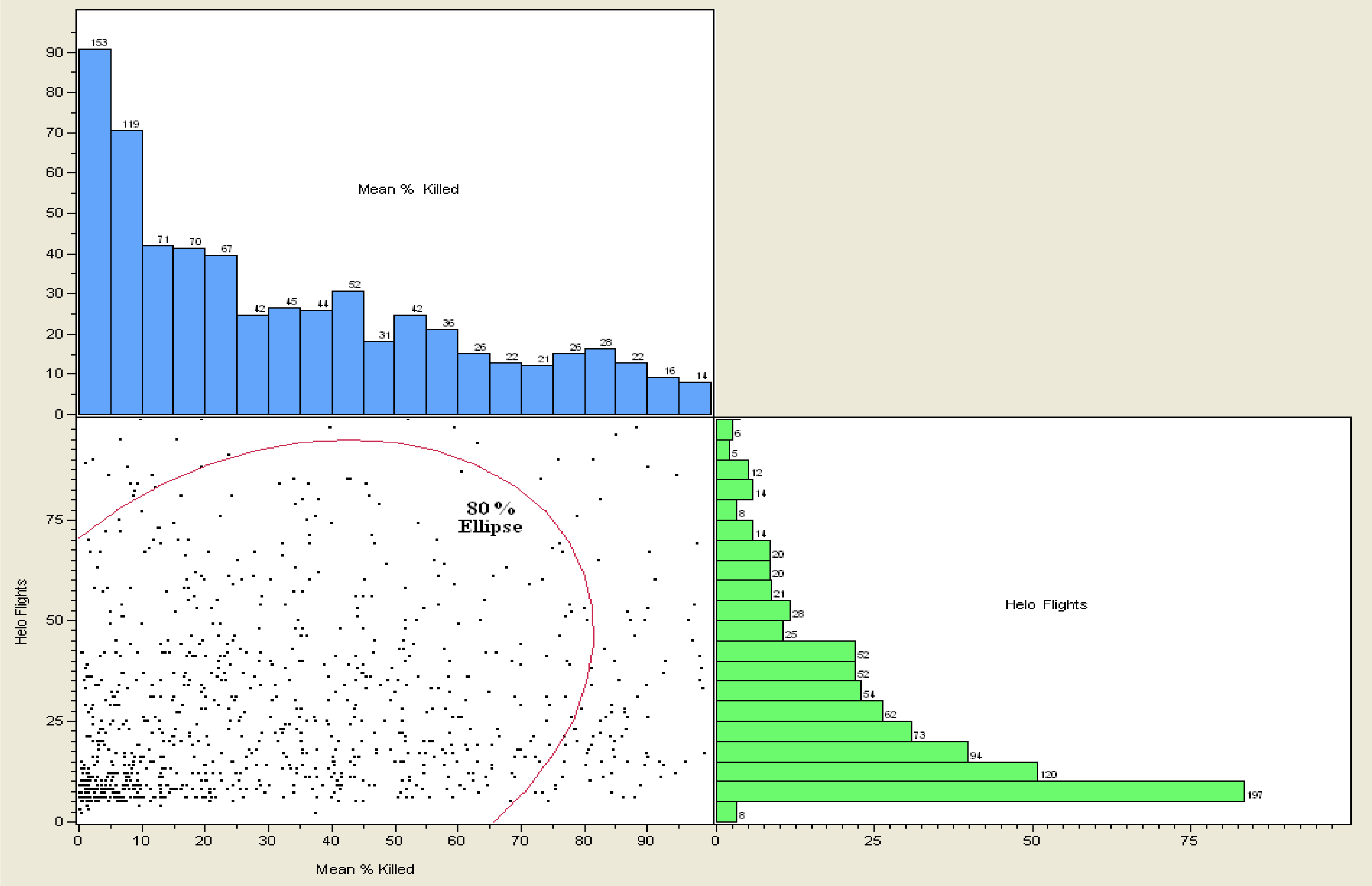


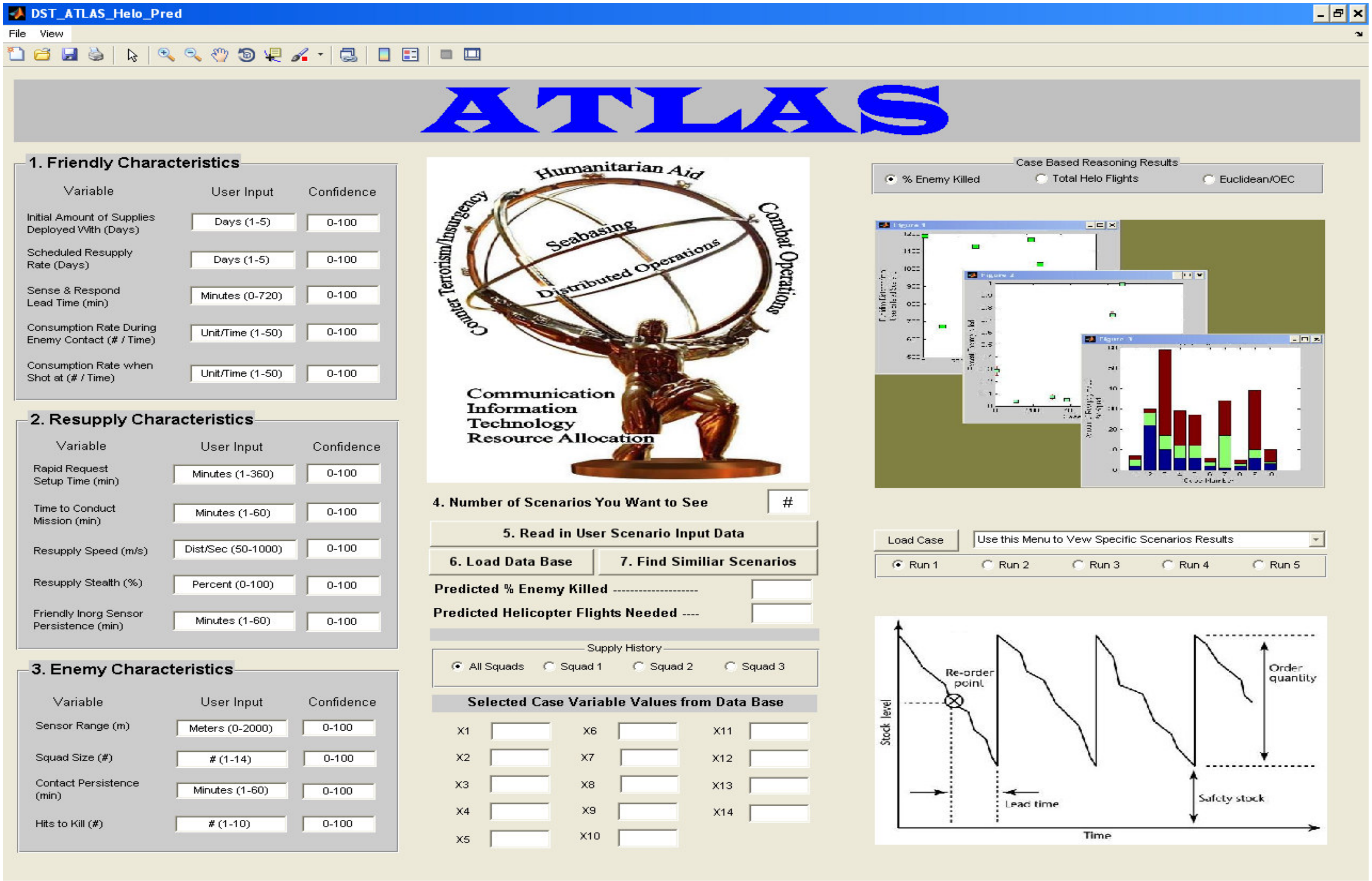
Figure 59: Plot of *Percent Enemy Killed* versus *Number of Resupply Flights*

Inclusion or exclusion of cases is accomplished by the user specifying the ellipse size and then selecting the cases within or outside the ellipse to be used for further analysis.

All of the plots discussed thus far enable the user to facilitate visualization, understanding, and manipulation of the data to gain a better insight to the design space. JMP and tools like it provide a wealth of visualization and exploration abilities to a well trained user who can manipulate the data created from the M&S environment within the application. Unfortunately the average DM at the operational level will likely find this process cumbersome and too time consuming to get any use out of the data in this format. For these DMs/logisticians at this level, quick answers along with some visualization in the form of graphs and options is more suitable. This type of information display will allow the DM to specify the input parameters and ultimately allow the logisticians to apply their expertise at the operational level to determine the most suitable COA based on the information available.

5.3.8.2 Decision Support Tool (DST): ATLAS GUI

The author felt a usable interface that incorporates some of the visualization tools and information from JMP and the MANA database would be very beneficial to the DM at the operational level. Computers are abundantly available to the operational level military logistician and applications like Excel are readily available throughout the military. The MANA DO platoon database was post processed and written to Excel because of this fact, but also because maintenance of this type of file should be easily accomplished at the user level if need be. To create the interface needed, the author turned to MatLab and its ability to create Graphical Use Interfaces (GUIs). The end result of this effort is shown in Figure 60. This tool allows a user to bypass complicated



1. Friendly Characteristics

Variable	User Input	Confidence
Initial Amount of Supplies Deployed With (Days)	Days (1-5)	0-100
Scheduled Resupply Rate (Days)	Days (1-5)	0-100
Sense & Respond Lead Time (min)	Minutes (0-720)	0-100
Consumption Rate During Enemy Contact (# / Time)	Unit/Time (1-50)	0-100
Consumption Rate when Shot at (# / Time)	Unit/Time (1-50)	0-100

2. Resupply Characteristics

Variable	User Input	Confidence
Rapid Request Setup Time (min)	Minutes (1-360)	0-100
Time to Conduct Mission (min)	Minutes (1-60)	0-100
Resupply Speed (m/s)	Dist/Sec (50-1000)	0-100
Resupply Stealth (%)	Percent (0-100)	0-100
Friendly Inorg Sensor Persistence (min)	Minutes (1-60)	0-100

3. Enemy Characteristics

Variable	User Input	Confidence
Sensor Range (m)	Meters (0-2000)	0-100
Squad Size (#)	# (1-14)	0-100
Contact Persistence (min)	Minutes (1-60)	0-100
Hits to Kill (#)	# (1-10)	0-100

4. Number of Scenarios You Want to See #

Predicted % Enemy Killed -------------------

Predicted Helicopter Flights Needed ----

Supply History

● All Squads ○ Squad 1 ○ Squad 2 ○ Squad 3

Selected Case Variable Values from Data Base

X1 X6 X11
X2 X7 X12
X3 X8 X13
X4 X9 X14
X5 X10

Case Based Reasoning Results

● % Enemy Killed ○ Total Helo Flights ○ Euclidean/OEC

● Run 1 ○ Run 2 ○ Run 3 ○ Run 4 ○ Run 5

Figure 60: ATLAS Decision Support Tool GUI

computer code and formulas that would preclude ease of use and functionality by the user unless they had extensive knowledge or ability to understand the underlying computer code. To begin using the ATLAS decision support tool (DST), the user enters information about the future scenario in blocks 1-3. These inputs reflect the 14 input variables used to represent the M&S environment and perturbed by the DOE in MANA. Once this information is entered the user can indicate how many *k Nearest Neighbor* (*kNN*) cases they would like to see that are near their notional or potential case in the entry block at Step 4. Using the *kNN* technique from CBR, the entered data is placed in vector format and used to search the database for the most similar cases to the one entered by the user. This is accomplished by calculating the Euclidean Distance as is also done in TOPSIS (Section 2.4.2). Step 5 reads in the users entered information and stores it in the GUI's memory workspace. Step 6 initiates the user to load the case based database from an Excel file predefined in the program code. Finally, Step 7 executes a series of calculations and executes a variety of plot functions that will now be available to the user.

It is important to note that the ATLAS DST calculates the *kNN* based on the enemy characteristics first and then finds the *kNN* from a subset of the database created by the first *kNN* calculation. This places emphasis on the enemy characteristic variables that ultimately will cause the greatest variability in the operational plan when they change. This will allow the user (logistician) to view the most relevant/similar cases in terms of the enemy situation and then assess the other variables that are relatively in the control of the military support structure and indicative of the DO platoon's capabilities.

The user now has available several pieces of visualization media. Under the Step 6 and Step 7 button is a prediction of the two MOE's based on the NN surrogate that was calculated earlier in JMP. In the top right are three plots activated by radio buttons that allow the user to see the cases identified by the *kNN* algorithm with the outcome of the mean of the *Percent Enemy Killed* based on the five simulations runs of each DOE case. The points also have error bars based on the standard deviation calculated from the averaging of these five runs. The next plot is a bar graph depicting the total number of *Helicopter Resupply Flights*. This plot consists of a bar chart that provides the total number of flights as well and a dissection of this total attributed to the three DO platoon squads. The third graph shows the Euclidean distance as well as a weighted OEC that takes into account the users uncertainty in the information and was described in detail in Section 5.3.6. These two distance metrics are plotted on the same plot to give the user the ability to see which cases might be best based on the *kNN* results and the variant OEC calculation expressed by their confidence in the data. Below these plots in the GUI is an area where the DM can select a case of interest from the database (based on the *kNN* results) and see a potential resupply timeline based on the MANA simulation runs. This allows the DM to not only see the number of flights, but also the distribution of the flights over the seven day period. This can be seen for each individual squad or all the squads for each of the five runs. This data is useful because it can show the DM the load the logistics support units in support of the DO platoon might expect over the course of the mission. Examples of these plots and data outputs will be shown in the application of a notional test case next.

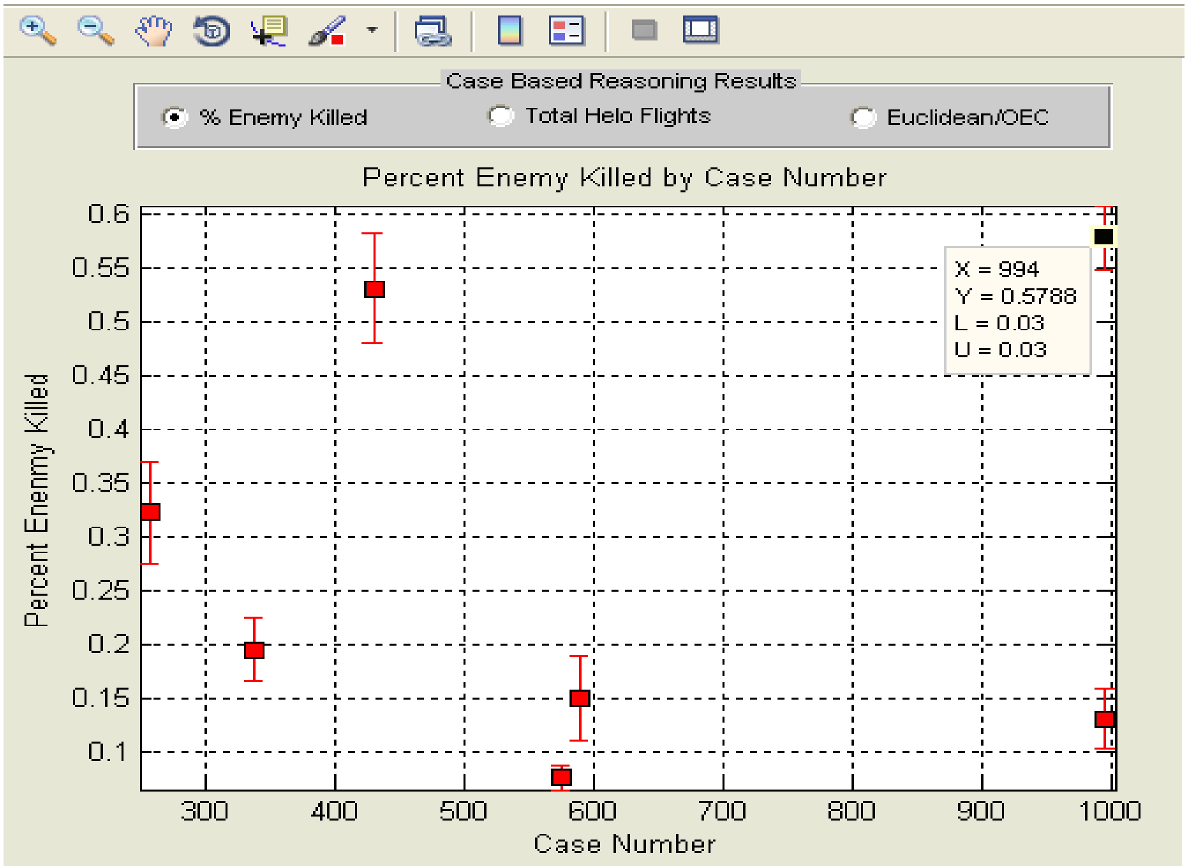
5.3.9 Notional Test Case for DST ATLAS

Using the settings in Table 17, a query for seven kNN cases was conducted. The ATLAS DST as stated, also incorporates the NN equation generated from JMP to give the user a prediction of the two MOEs of interest, *Percent Enemy Killed* and *Helicopter Resupply Flights.* The results are calculated within MatLab code that can be viewed in Appendix C 2 & 3. The predicted MOE outcomes for these settings are 86.93% of the enemy killed, with 39 (38.49 rounded up) helicopter flights required. If the user is satisfied with this prediction they can move on or they can modify the inputs to see if an improvement in the MOE's can be achieved. The next part of the GUI provides visualization and case based information on the *kNN* cases extracted from the database. The first visualization section is located in the upper right hand corner of the GUI and provides three different plots activated by radio buttons described earlier.

Table 17: DST Exploration Notional Inputs

Variable (units)	Value	Confidence	Variable (units)	Value	Confidence
X1: DO Squad Days of Supply (Days)	**3**	**100**	X8: Resupply Speed (knots)	**110**	**75**
X2: Scheduled Resupply Rate (Days)	**2**	**90**	X9: Resupply Stealth (%)	**100**	**75**
X3: Sense and Respond Lead Time (min)	**150**	**75**	X10: Friendly Inorganic Sensor Persistence (min)	**30**	**80**
X4: Consumption Rate when in Enemy Contact (#/t)	**25**	**65**	X11: Enemy Sensor Range (meters)	**300**	**80**
X5: Consumption Rate when Shot at (#/t)	**35**	**75**	X12: Enemy Squad Size (#)	**5**	**75**
X6: Rapid Request Setup Time (min)	**45**	**75**	X13: Contact Persistence Enemy (min)	**15**	**65**
X7: Time to Unload Resupply Helicopter (min)	**15**	**90**	X14: Hits to Kill Enemy (#)	**2**	**75**

The first plot result for the seven *kNN* cases is shown in Figure 61. Also incorporated into the GUI and shown above the first plot, is a tool bar that enables the



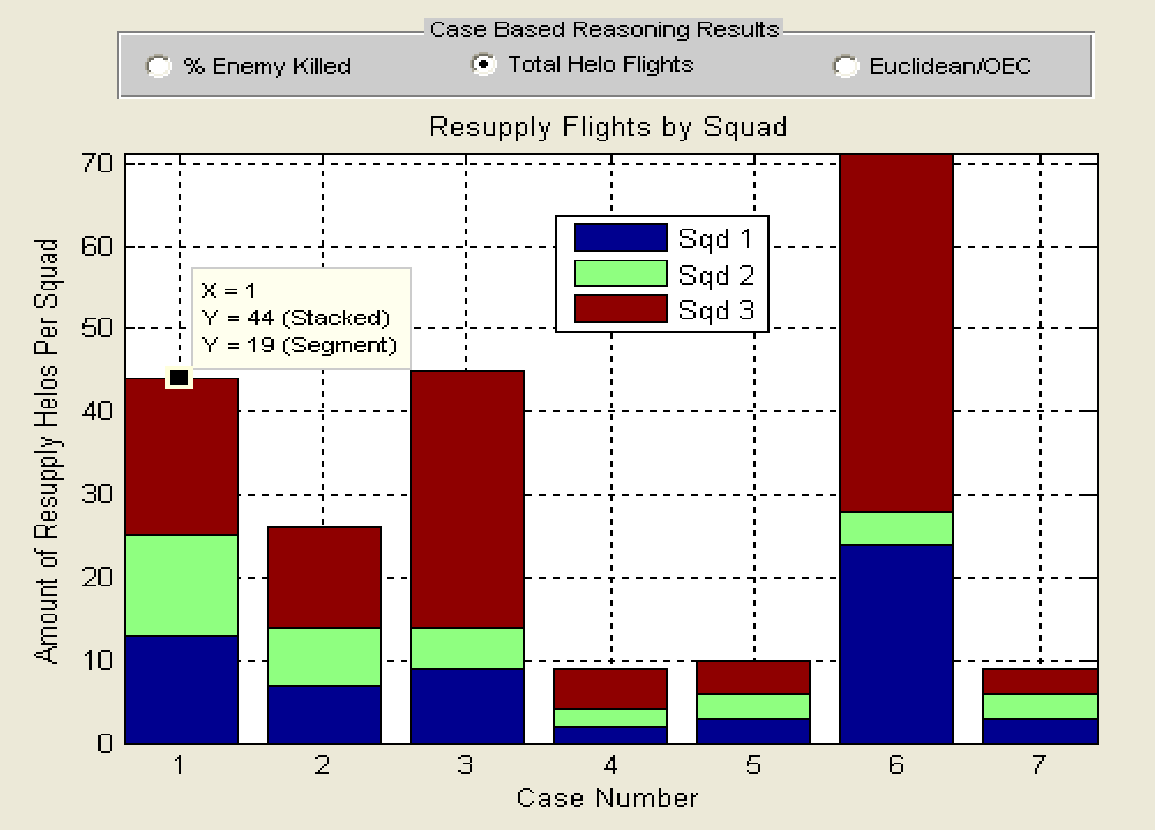
Case Based Reasoning Results

● % Enemy Killed ○ Total Helo Flights ○ Euclidean/OEC

Figure 61: Percent Enemy Killed by Case Number from DST ATLAS

user to manipulate these sets of plots to get detailed information from the underlying calculations or specific case characteristics. This is be achieved by selecting the icon, , and then selecting the data point the user wants more information about. An example of this capability is also shown in Figure 61 where the author selected the highest value of percent enemy killed for this query. The result is a data box with X = 995,Y = 0.5788, and L and U = 0.03. This corresponds to case number 995, with an average percent enemy killed of 57.8 % and a standard deviation, based on the five runs on the input variables, of 3 percent. The user can also zoom, rotate, pan, and draw on the graph, or they can apply a legend or inset a color bar for the data. This graph tool bar can be utilized on any of the plots within the DST ATLAS GUI. As can be seen, the MOE values on the plot do not look to reflect the high yielding percentage predicted by the NN surrogate incorporated into the GUI. This is the result of the selected cases not exactly matching the query case. Now the user must apply analysis and expertise to digest the information being presented to understand the implications of these nearest neighbor cases. Before that task is attempted, a visualization of the other plots is prudent.

The second plot displays the total number of helicopter flights for each of the seven cases and is shown in Figure 62. These cases reflect flights that range from as small as 9 to as much as 71. The bar chart shows the total number of flights as well as the portions of the total for each of the three DO platoon squads as indicated by the legend. The two command elements are excluded in these plots because their needs were determined to be so minimal that it is assumed that resupply activities for them will occur



Case Based Reasoning Results

○ % Enemy Killed ● Total Helo Flights ○ Euclidean/OEC

Figure 62: Total Number of Helicopter Flights by Case from DST ATLAS

in route to the larger squads. It is important to note that the number of helicopters is an average taken across the five replications for each DOE case. This chart currently shows there are two, possibly three cases that are in the vicinity of number of helicopter flights predicted by the NN surrogate.

The last plot of this section portrays the distance metrics in shown in Figure 63. Here the user can compare the strict Euclidean Distance versus the OEC variant equation

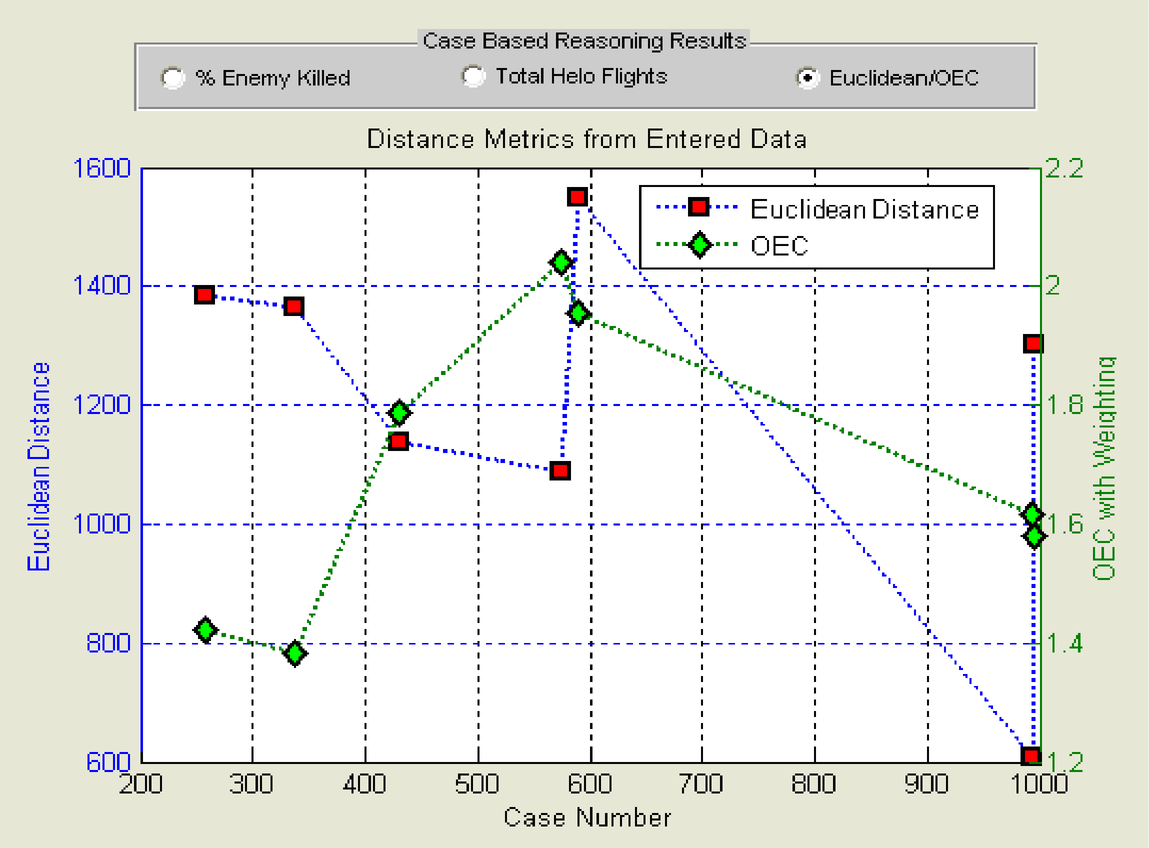


Figure 63: Distance Metrics (Euclidean and OEC) from DST ATLAS

that is calculated with the unique weighting scale that incorporates the fuzzy inference system scaling algorithms discussed earlier and executed by MatLab. Each metric result is on a different scale with the Euclidean distance plotted along the Y1 axis and the OEC value plotted along the Y2 axis. A synopsis of the *kNN* cases retrieved with their specific variable values, associated MOE values, and calculated metrics is presented in Table 18.

Table 18: Data for Seven kNN Query in DST ATLAS

Variable	**Query**	**Database Cases**						
		994	**575**	**430**	**995**	**338**	**257**	**590**
DO Squads Days of Supply	3 (100)	3.15	3.23	2.95	2.93	2.85	2.56	2.98
Scheduled Resupply Rate	2 (90)	1.95	1.89	2.31	1.76	2.35	2.17	2.46
Sense&Respond Lead Time	150 (75)	112	482	369	687	108.5	157.5	130.5
Consume: Enemy Contact	25 (65)	48	49	17	26	9	23	34
Consume Rate: Shot at	35 (75)	32	45	42	19	29	20	0
Rapid Request Setup Time	45 (75)	50	176	246	112.5	87.5	5	133
Time for Supply Mission	15 (90)	55	55	54	3	27	50	57.5
Resupply Speed	110 (75)	497	468	430	85	908	334	869
Resupply Stealth	100 (75)	14	41	95	13	82	58	26
Friendly Sensor Persistence	30 (80)	40	9.5	3	43	4	45.5	24
Enemy Sensor Range	300 (80)	480	480	720	630	120	390	510
Enemy Squad Size	5 (75)	8	6	8	12	3	2	1
Contact Persistence Enemy	15 (65)	22	23.5	9	13	10	11.5	23
Enemy Hits to Kill	2 (75)	2	7	2	9	4	9	9
% Kill Killed	▨	57.88	7.58	53.13	13.04	1.95	32.25	15.00
Standard Deviation	▨	3.0	1.12	5.13	2.8	2.92	4.79	3.95
Number of Resupply Flights	▨	44	26	45	9	10	71	7
Euclidean Distance	▨	608.29	1088.25	1137.58	1301.22	1365.94	1385.70	1549.13
OEC Variant	▨	1.61	2.04	1.79	1.58	1.38	1.42	1.96

The first plot from the ATLAS DST GUI showed the user that at two cases 994 and 430 had the highest values, 57.9 % and 53.1 % respectively. A review of these cases should be conducted. The second plot shows that these two cases in addition to 575 are good candidates for further review based on the number of resupply flights that are similar in number compared to the predicted value by the NN surrogate. The third plot shows that these three cases have the smallest Euclidean distance and this further validates the need to delve deeper into the specific characteristics of these three cases. While this direction is the first recourse, a look at the OEC plot shows the three cases with the closest OEC to one are 995, 338, and 257. These cases may also need to be investigated. A Look at case 994 that has about a 58 % enemy killed percentage has 44 flights. Figure 64 shows the actual resupply schedule of the logistics helicopters by day

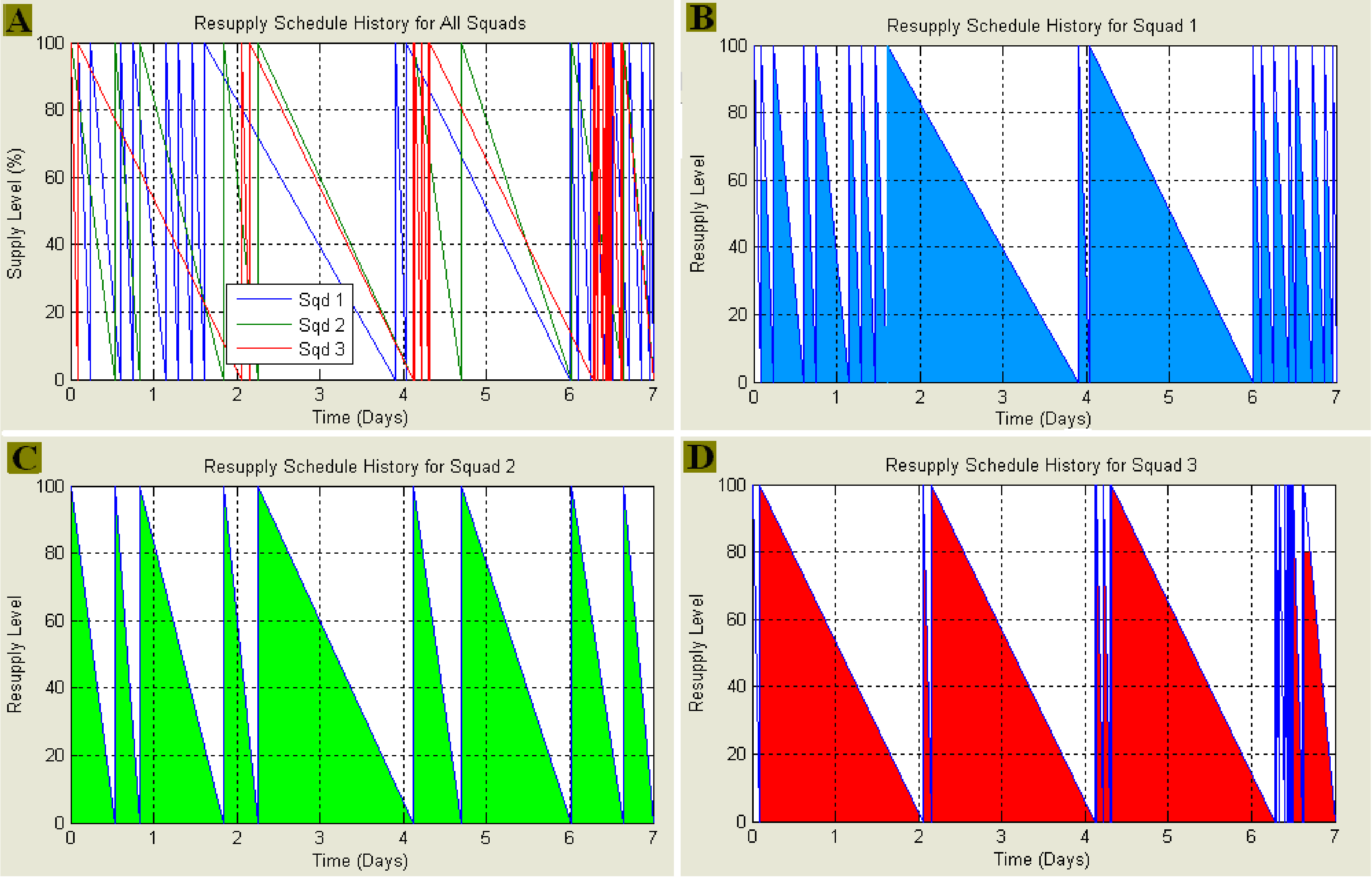


Figure 64: Resupply Schedule for Case 994 - Total and by Squad

and a linear forecast of usage by each of the squads for this case. Theses plots are individually generated in the window in the lower right hand corner of the GUI and are based on the case, run, and squad selected by the user. Plot A shows the combination of all three squads and although difficult to read shows the DM that there is a trend of heavy enemy contact in the beginning of the operation, with a lull in the middle and then an increase in enemy contact toward the end. To get a better understanding of individual squad requirements the DM can look at plots B, C, and D. For Squad 1, the Northern most squad, the DM can see there is quite a bit of activity, which could suggest the enemy is trying to use a route that was either previously safe or they felt they could easily traverse. For Squad 2, plot C shows a relatively steady usage of supplies and exposure to enemy combatants. The steady contact could be the result of enemy combatants' vicinity the Northern passage who are now heading South due to contact with the first squad. Finally, plot D shows resupply occurring for Squad 3 just about every 2 days and on schedule. This tells the logistician that the planned resupply of every two days was suitable for this squad, but the other plots reveal that a revaluation for Squads 1 and 2 in terms of their supply forecasts may needs to be conducted.

The ATLAS DST at this point has enlightened the DM in several ways. The first was by providing useful information in the way of possible logistics load on aerial resupply assets. It also allows the user to utilize expertise in looking at a wide variety of characteristics to assess possible enemy actions. Other things not necessarily incorporated into this tool but can be performed at the lower level by the HITL, are terrain analysis, enemy trends when attempting to cross the border, and determination of intelligence accuracy, past and future. Each of these activities can be incorporated by the user while executing the ATLAS DST. While this is beneficial a look at the other cases is warranted and should be useful in surmising an overall logistical support picture and be beneficial in determining a potential load forecast and forecast for resources needed for the seven day operation on the border.

The next case, 430, has the resulting plots for the helicopter resupply schedule shown in Figure 65. This case, although similar in results with case 994, only has a value of 53% enemy killed and experienced 45 resupply flights. This plot shows a completely different result in the frequency of flights. Enemy squad size is the same, but the enemy's ability to sense the DO platoon is 1.5 times greater and beyond the effective range of small arms weapons in the DO platoon's arsenal. A look at other factors also reveals that *Sense and Respond Lead Time* is 2.46 times greater than the user's query and almost three times as long as in case 994. This will most certainly have an effect on the delivery of supplies and tells the DM how important keeping this variable minimized can be. Case 994 had a lead time of 112 minutes and is likely a good indicator of the load on the aerial resupply assets. With the low percentage of enemy killed in these two cases, the DM can also start to explore what other assets are available that can increase long range visibility for the DO platoon. Unmanned Aerial Vehicles (UAV's) are relatively prevalent in today's military and could be leveraged to provide intelligence to the DO platoon in its area of operation. The DM may also want to look at reducing the area covered by one platoon and depending on the mission intent and criticality of the outcome, whether a different force architecture is warranted. Exploration of these cases should now be providing the logistician with a foundation to start determining the support architecture that will be needed to support the DO platoon and its mission.

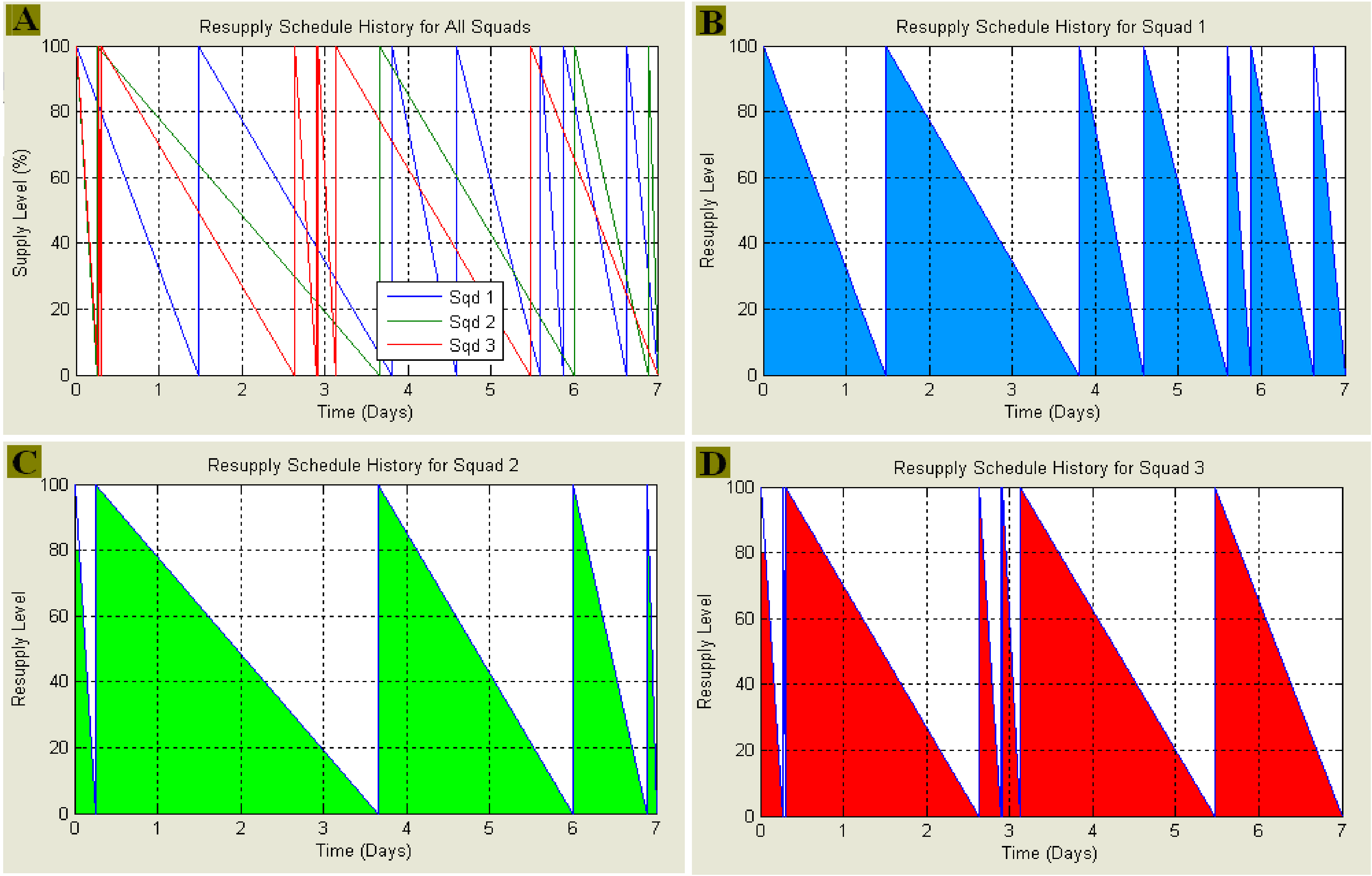


Figure 65: Resupply Schedule for Case 430 - Total and by Squad

The last case to be examined in detail is case 575. The results for the helicopter flights are shown in Figure 66. The four plots show a relatively even delivery rate for each of the squads, which is a plus for reducing the variability in the number of flights and when they will occur, but the MOE for percent enemy killed is very low at 7.58 %. Further analysis points to two variables that are the likely culprits. The first is X14 or the number of hits to kill an enemy agent. On a scale of 1 to 10 this number is at a 7, which means this enemy is highly trained and very difficult to kill. Also the *Sense and Respond Lead Time* is 3.2 times longer than what is being expected by the logistician. While the case as it stands shows no viability in how the commander would approach solving the problem it does highlight again the importance of having adequate sense and respond capabilities as well as knowing the ability of your enemy so that the size and architecture of the force is suitable to engage the enemy in their “known” state.

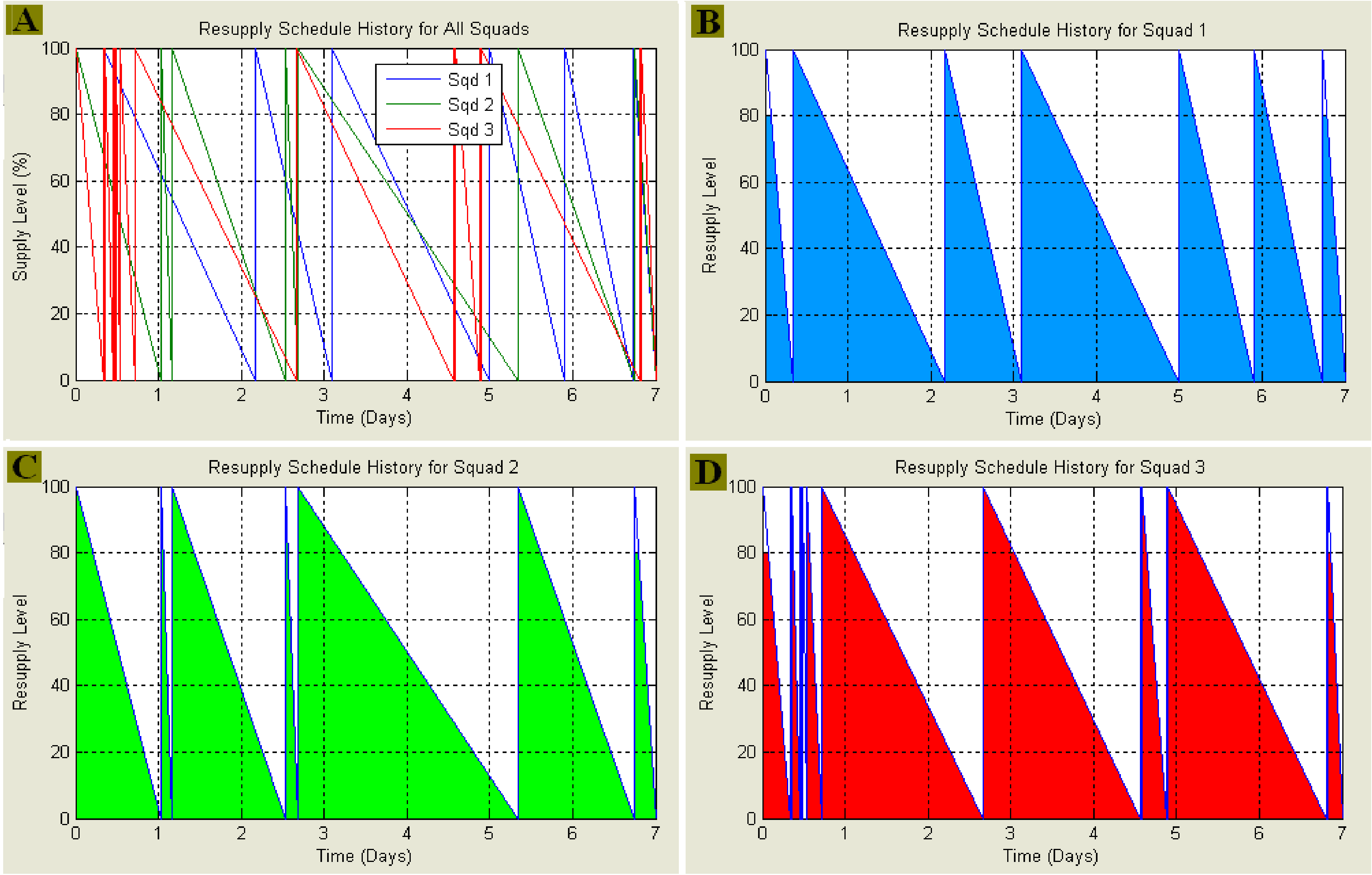
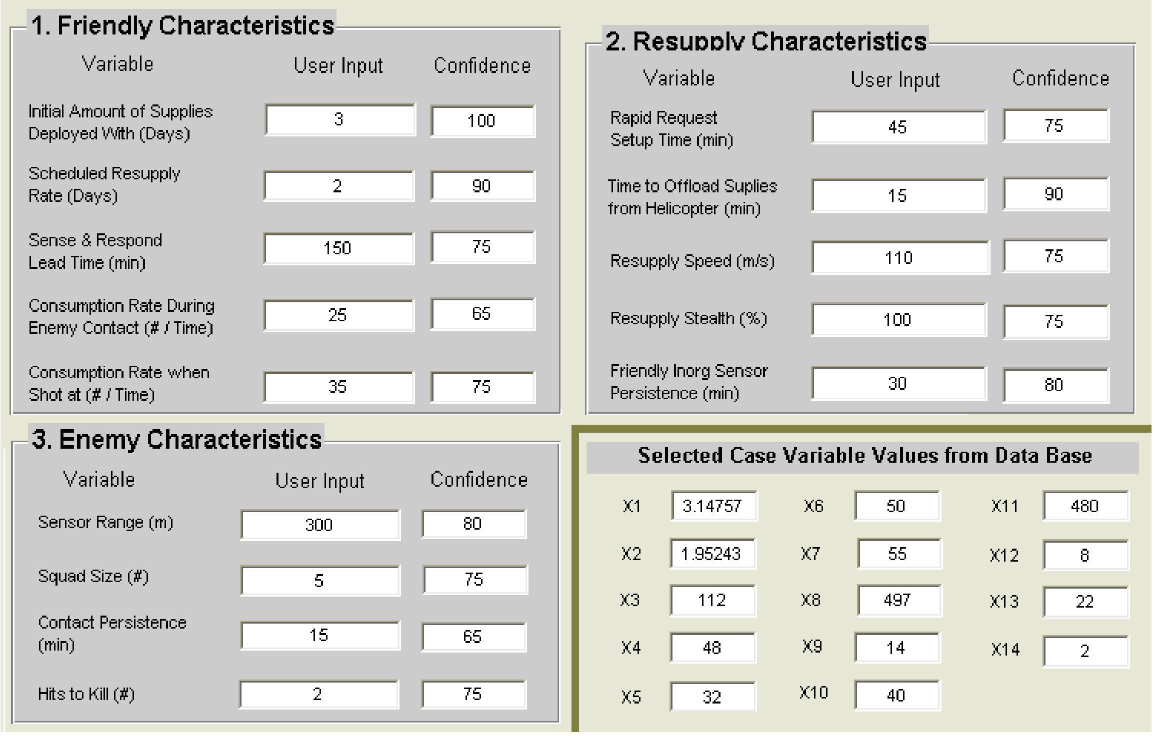


Figure 66: Resupply Schedule for Case 575 - Total and by Squad

Comparison of the cases is beneficial because the operational level user can start to see why a particular case succeeded or failed by seeing the variables side by side. Analysis of this nature is easily performed in the ATLAS DST GUI because when the user executes the *Load Cases* button that displays the series of plots shown in Figure 64 through Figure 66. This button also loads the selected case's 14 variable values from the database as shown in Figure 67. This prevents the user from having to go into the database and conduct a tedious search for a retrieved case. This capability enables very quick trade studies of the retrieved case selections to that of the query and allows the user to apply operational expertise in finding a solution for the current environment being planned for.



1. Friendly Characteristics

Variable	User Input	Confidence
Initial Amount of Supplies Deployed With (Days)	3	100
Scheduled Resupply Rate (Days)	2	90
Sense & Respond Lead Time (min)	150	75
Consumption Rate During Enemy Contact (# / Time)	25	65
Consumption Rate when Shot at (# / Time)	35	75

2. Resupply Characteristics

Variable	User Input	Confidence
Rapid Request Setup Time (min)	45	75
Time to Offload Suplies from Helicopter (min)	15	90
Resupply Speed (m/s)	110	75
Resupply Stealth (%)	100	75
Friendly Inorg Sensor Persistence (min)	30	80

3. Enemy Characteristics

Variable	User Input	Confidence
Sensor Range (m)	300	80
Squad Size (#)	5	75
Contact Persistence (min)	15	65
Hits to Kill (#)	2	75

Selected Case Variable Values from Data Base

X1	3.14757	X6	50	X11	480
X2	1.95243	X7	55	X12	8
X3	112	X8	497	X13	22
X4	48	X9	14	X14	2
X5	32	X10	40		

Figure 67: Variable Comparison Example for Case 994 and User Query

A look at the other three cases reflected by the OEC plot (995, 338, and 257) reveals some large and small differences between the query input variables. The small differences seem to occur for the variables where high confidence was a trait and large difference appears to occur where confidence was moderate to low. The confidence in X1, X2, and X7 were high and resulted in a close convergence of matching for the query data where as low confidence (75 and below) resulted in a varied spread across the ranges for those particular variable. The ATLAS method places less emphasis on the low confidence variables because from a holistic view they are noisy and are likely to change. Consequently, the OEC variant seeks to search amongst all of the retrieved cases for the cases that should provide further insight to trends, or relationships not seen earlier. A look at these three cases does reveal that X3 and X14, *Sense and Respond Lead Time* and the *Number of Hits to Kill an Enemy* respectively, do seem to have a large role in the *Percent Enemy Killed* MOE. This is interesting because if the reader looks back at the Pareto Plot in Figure 54 for this MOE, X14 is the number one variable of influence and X3 is the seventh. A look at the Helicopter Resupply flights MOE shows X3 as the third highest variable of impact on this metric, but X14 is the last. This highlights the criticality of a decision support tool that needs to be able to visually allow observation of outcomes for a user in a quick and easy manner. This also reinforces the need for relevant and accurate information from complex, nonlinear, and integrated environments.

5.3.10 Step 9: Determine COA for Architecture Selection

The final step of the ATLAS method is for the user to select a COA and viable logistics architecture based on the information garnered by executing the ATLAS method and its tools. The method guides the user through information based on simulated data of similar situations to a probable outcome in terms of logistics requirements to sustain the DO platoon for its mission. This information incorporates certainty in the information at the time of the query as well as intelligence about the enemy and known knowledge about friendly forces. The products from the ATLAS DST GUI give the operation level planner an idea of the logistical need and hence the potential load that could be experienced on resupply assets. These assets whether helicopters, aircraft, trucks, etc. are commodities that needs to be managed as well as possible to prevent a breakdown in the logistics delivery system. While some planners will have the freedom to create their architecture via requests for assets, others will be limited to what they have organic to their unit or at their disposal. Fortunately this is where the HITL and the expertise of the logistician as the DM can be leveraged to insure flexibility and robustness of the overall force architecture is achieved.

The ATLAS method and supporting tools is not likely to provide the exact answer for the military DM seeking a solution to the complex operational environment they are facing. While not the 100 % solution, the ATLAS method and the decision support tool encompassing the processes from this method do help to provide better insight into the potential outcomes for the DM. The method aids in narrowing down the spectrum of the similar situations so that decisions can be made quickly based on operational data, experience, and intelligence - not just the intuition of the DM. The capability of this method and the tools developed also allow a DM to explore the design space rapidly to limit risk and adverse outcomes. Insights gained via the visualization products produced bring knowledge and options into the decision making process and should minimize the time it takes to conduct military decision making and planning processes.

The data and plots provided by the ATLAS DST are the first steps in selecting a suitable and capable logistics architecture for supporting the DO platoon for their border deterrence mission. Case 994 appears to be a good starting point for development of this architecture. Limitations of the modeling software preclude dynamically changing the resupply amounts to the squad during the simulation runs so it is assumed that resupply packages for ammunition, food, and water will be setup for the standard "3 days of supply" quantity outlined in Table 8 and shown here again for reference. Food and water weigh 736 and 4400 pounds respectively.[1] This equates to a total weight of 9305 pounds for three days of supply for the platoon, this is approximately 3100 pounds of supplies per squad. Expressing this package's characteristics in detail now allows the LCE of the MEU to start prepackaging standardized support packages for delivery either at a scheduled resupply point or for a rapid request for resupply initiated from the field

DODIC	NOMENCLATURE	WPN TAMCN	ROUNDS PER WEAPON	TOTAL ROUNDS	TOTAL WEIGHT	TOTAL CUBE
A059	CTG, 5.56MM BALL	E1441	350	12,250	502	12
A063	CTG, 5.56MM TRACER	E1441	10	350	17	1
A064	CTG, 5.56MM 4 & 1 LINKED	E0960	800	7,200	439	14
A131	CTG, 7.62MM 4 & 1 LINKED	E0989	800	2,400	242	5
A363	CTG, 9MM BALL	E1250	30	60	4	0
A576	CTG, CAL .50 4 & 1 LINKED	E0984	400	2,000	790	12
B504	CTG, 40MM GREEN STAR PARACHUTE	E0892	0.3	5	5	0
B505	CTG, 40MM RED STAR PARACHUTE	E0892	0.3	5	5	0
B506	CTG, 40MM RED SMOKE GROUND	E0892	0.25	5	5	0
B508	CTG, 40MM GREEN SMOKE GROUND	E0892	0.5	5	5	0
B509	CTG, 40MM YELLOW SMOKE GROUND	E0892	0.25	5	5	0
B535	CTG, 40MM WHITE STAR PARACHUTE	E0892	2	18	19	0
B542	CTG, 40MM HEDP LINKED FOR MK19	E0994	288	1,152	864	23
B546	CTG, 40MM HEDP FOR M79/M203	E0892	18	162	119	3
G881	GRENADE, HAND FRAGMENTATION	INDIV	1 PER INDIVIDUAL	44	75	2
G900	GRENADE, HAND INCENDIARY TH3	INDIV	1 PER ITV	11	32	1
G930	GRENADE, HAND SMOKE HC	INDIV	1 PER SQUAD OR HQ	5	13	0
G940	GRENADE, HAND SMOKE GREEN	INDIV	1 PER SQUAD OR HQ	5	13	0
G945	GRENADE, HAND SMOKE YELLOW	INDIV	1 PER SQUAD OR HQ	5	13	0
G950	GRENADE, HAND SMOKE RED	INDIV	1 PER SQUAD OR HQ	5	13	0
G955	GRENADE, HAND SMOKE VIOLET	INDIV	1 PER SQUAD OR HQ	5	13	0
L306	SIGNAL, ILLUM GROUND RSC	INDIV	1 PER SQUAD OR HQ	5	8	0
L307	SIGNAL, ILLUM GROUND WSC	INDIV	1 PER SQUAD OR HQ	5	8	0
L311	SIGNAL, ILLUM GROUND RSP	INDIV	1 PER SQUAD OR HQ	5	8	0
L312	SIGNAL, ILLUM GROUND WSP	INDIV	1 PER SQUAD OR HQ	5	8	0
L314	SIGNAL, ILLUM GROUND GSC	INDIV	1 PER SQUAD OR HQ	5	8	0
L323	SIGNAL, SMOKE GROUND RP	INDIV	1 PER SQUAD OR HQ	5	8	0
L324	SIGNAL, SMOKE GROUND GP	INDIV	1 PER SQUAD OR HQ	5	8	0
L495	FLARE, SURFACE TRIP	INDIV	1 PER SQUAD OR HQ	5	9	0
				Total	3,258	77

---

[1] $5.58\frac{lbs}{day} * 3\ \ days * 44\ \ men = 736\ \ lbs$; $8.33\frac{lbs}{gal} * 4\frac{gal}{day} * 3\ \ days * 44\ \ men = 4398\ \ lbs$

environment. This also enables the customer or DO platoon the ability to plan, coordinate, and execute their mission based on the package characteristics. This does not preclude the adding or taking away of commodities from the logistics packages, but it does allow for some efficiency to be realized in terms of reducing reaction time and it helps to make the process of getting logistics to the right people, to the right place, and at the right time a less volatile task by creating a baseline support package.

A three day supply consisting of food, ammunition, and water for the entire platoon comes to 9305 pounds. Unfortunately this does not include the necessities to sustain the vehicles organic to the platoon. Bain in [8] gives a suitable approximation of a total weight of these items consisting of Class II, III, and V at 628 pounds per vehicle for three days. With 11 vehicles assigned to the entire platoon the total weight to support the vehicles equates to 6908 pounds; consequently, to supply a platoon fully with the theoretical "3 days of supply" amounts to 16214 pounds of material. Table 19 displays the breakdown for each element of the DO platoon for resupply needs based on the "3 days of supply" quantities. Referring to Table 11, a single aircraft is able to resupply either the entire platoon or squads individually based on their needs. An efficient use of the resource would be to resupply everyone at once, but as the analysis from the ATLAS DST has shown this is not necessarily the need nor is it a feasible COA from a logistics perspective, as the DO platoon theoretically is not the only unit supported by the MEU.

Table 19: Resupply Requirements per DO Platoon Element for Three Days

Element	Soldiers		Vehicle		Total
Command A or B	4	846 lbs	1	628 lbs	1474 lbs
Squad 1, 2, or 3	12	2538 lbs	3	1884 lbs	4422 lbs

This highlights the arduous task facing the military logistician; how to forecast logistical requirements and needs for ambiguous and dynamic operational environments.

Fortunately the ATLAS DST can give some insight. Referring to Figure 64, the logistician can begin to plan for the mission by using the outcomes reflected via the MANA ABM. Table 20 shows a possible supply need in terms of the number of three days of supply packages from the beginning of the mission till the end of the mission based on the ATLAS DST visualization products.

Table 20: Notional Need for DO Platoon 3 Day Resupply Packages

Element	Day 1 to 2	Day 2+ to 4	Day 4+ to 6	Day 6+ to EOM
Squad 1	8 (20304 lbs)	1 (2538 lbs)	2 (5076 lbs)	7 (17766 lbs)
Squad 2	4 (10152 lbs)	2 (5076 lbs)	2 (5076 lbs)	4 (10152 lbs)
Squad 3	1 (2538 lbs)	1 (2538 lbs)	3 (7614 lbs)	2 (5076 lbs)
Total Packages	13	4	7	13

The sum of the total projected need is approximately 36 three day packages. The ATLAS DST if the reader recalls predicted 39 packages via the NN surrogate, and Case 994 which this estimates is based off of utilized 44 packages.

With this information the logistician can now start planning for the mission. The forecasted need is for 36 standard "3 days of supply" packages for the course of the mission. While this is the forecasted need, the logistician will likely need to retain some flexibility in the final number because the environment is evolving as he/she sets the logistics plan in motion. With this forecast, a breakdown of the previously described tables in terms of Class I, II, III, V, etc can be established and inventory of these items can either be confirmed or immediately placed on order by the logistician through the military supply system. This information also provides the LCE with guidance as to their expected requirements in terms of logistic packages and enables them to begin prefabricating at least the initial 13 "3 day supply" packages forecasted for the deployment phase of the operation. Table 20 shows a forecast quantity per squad that places heavy emphasis on Squad 1. While this is how the model in MANA portrayed the outcome, this is not necessarily going to be the result in reality. This is why the HITL is critical to the planning and execution of military operations. The HITL understands that any of the squads could face initial heavy contact, therefore one might decide to send each of the squads out with their basic load of three days plus four more “3 days of supply” packages to cache at their base of operations on the border. As the mission evolves the logistician can monitor the activity from platoon reports or other feedback sensors to adjust the delivery schedule of the resupply aircraft to meet specific needs. This decision would not only allow some buffer for each of the squads, but would also achieve the intent of keeping the enemy completely unaware of resupply activities as they would be irregular and sporadic based on the unique forecast determined via the ATLAS DST. This aspect of the planning and architecture selection process fulfills the desire by military leaders expressed earlier to create confusion for the enemy via asymmetric actions. It also should, among other things, potentially increase the level of security resupply aircraft experienced since the enemy will not be able to track the sporadic resupply schedule.

The information from the ATLAS DST allows the logistician to coordinate with the ACE for support in the form of aircraft and crew. The information garnered and processed from the ATLAS DST can be used by the ACE to help determine the likely flight load/schedule. The ACE generally has to consider the number of mission capable aircraft, the number of pilots and crew they have for duty to create this plan, now they should be able to plan and coordinate other tasks such as maintenance and external support for other units. While the schedule is not necessarily set in stone, getting this forecast to the ACE and LCE allows each of them to prepare and plan, thus providing increased flexibility to the MEU and its supported units. If the plan is not achievable by the ACE and or LCE with their current resources, that realization should be arrived at sooner and allow for commanders and DM's more time to resolve potential issues. The ATLAS DST has and can be utilized to conduct mission analysis to bring critical situational characteristics to light for the DM's. This should result in increased flexibility and robustness of the MEU and its subordinate units when faced with vague and evolving small unit operations.

5.3 Chapter Summary

In this chapter the ATLAS method and its nine step process was applied to provide a decision support framework for determining a logistics architecture in support of a scaled military operation conducted by a DO platoon. The first two steps required the problem to be defined and aided in framing the critical components that enabled the third step. This step was facilitated by an agent based model representing the operational environment. Verification of the model was conducted to insure the model was working as intended accomplished Step 4. Step 5 was arguably to most difficult step in the methodology simply because of the shear volume in data and coding necessary to parse the data from the M&S software environment and reports. Step 6 applied fuzzy logic and linguistics through the use of off the shelf software, MatLab. This application allowed for the determination of weights to be applied for evaluation of the cases selected from the database. Step 7 and 8 also used off the shelf software (JMP) to facilitate visualization of the data. A custom GUI DST was created to allow the operational level user to conduct useful design space exploration by facilitating quick extraction of pertinent data that aided in the selection of a robust logistics plan for the scenario presented.

6 CONCLUSIONS AND CLOSING REMARKS

The problem facing the military joint logistics system is on a massive scale and will require some very innovative techniques, methods, and application of emerging technologies to solve. The areas of research concerning fuzzy logic, neural network applications, and agent based modeling are enjoying renewed vigor in the design community for their individual and combined abilities to model and simulate real world effects and problems. Application of new methods and modifications to existing methods with new concepts is occurring constantly and is sure to be the subject of much research and discussion as long as designers and engineers continue to solve increasingly complex and nonlinear problems. Technology advancements, whether it is in the area of increased speed, capability, capacity in the realm of computing, or if it is the implementation of once cost prohibitive technologies such as RFID tags, engineers will need to continue to try and solve problems with more and more efficiency.

Ideally anyone involved in the logistics realm would like to assume stable demand. This reduces the variability that is complicated to incorporate into forecasting models. While this simplifies the analysis of the environment, it is often inflexible to change and can be catastrophic in the face of volatility. The military operational arena is a dynamic system that prevents 100 % accurate predictions or forecasting in many areas, but that does not mean it should be ignored. Missiles and bombs will continue to be used in combat operations, but putting soldiers on the ground will likely always be a necessity to have any ability, post conflict, to conduct follow on efforts. The future holds many uncertainties in what the immediate and long term military operational landscape will be or require. Operations that require response of only a small military element are thought to be the future environment for military. Many organizations in the military structure have indentified logistical support for this type of environment as either inadequate or nonexistent. This work has shown that applications of knowledge, inherent or gained through modeling and simulation, cognitive reasoning methods, and useful visualization techniques, can be beneficial in providing a foundation for decision makers to select courses of action and viable support plans. This information helps the logistician formulate, plan, and execute the peripheral tasks needed to create suitable and hopefully flexible logistics architectures for small unit operations; in the end, providing the military an option that can aid in getting the right resources, to the right people, at the right time.

6.1 Assessing Hypothesis Statements

Four hypotheses were expressed in Chapter 3 and consequently should be evaluated as the culmination of this work is reached. The first statement reflected the overarching intent of this work, how to provide a decision support network for military logistics decision makers and is stated below:

> *Any method formulated to provide decision support system for determining military logistics architectures should incorporate a combination of historical data, cognitive reasoning, and variation in the operational environment.*

Contribution: The ATLAS method provides a methodical approach that incorporates applicable quantitative information in concert with user assessments and expertise (qualitative information) to determine suitable courses of action via an easy to use GUI. In the past DM had to rely almost solely on inherent knowledge with no way to access relevant and potentially useful information or data to help further guide their decisions. The ATLAS method brings knowledge in sooner and fills this gap.

While the ATLAS method certainly is not the only way to provide decision support as evidence by the morphological matrix in Table 4, it is one option. This method utilizes data from a representative modeling and simulation environment, along with user assessments and expertise to help formulate a logistics architecture and plan based on quantitative and qualitative information. In the past DM had to rely almost solely on inherent knowledge with no way to access useful information or data to guide their decisions. This method provides not only useful quantitative information, but allows the user to apply their expertise to holistically consider the environment and adjust plans to meet the determined need. By not removing the human in the loop, this method seeks to enhance the flexibility already prevalent in the military environment by filtering the large amounts of data present and organizing it in such a way that military DM's can more quickly and efficiently perform their tasks.

The second hypothesis was developed out of the desire to facilitate a balance between the quantitative and qualitative nature of a decision makers environment.

> *Historical or simulation data that provide low level characteristics should be used to quickly filter solutions for the decision maker through data based management techniques.*

> Contribution: Applying operational (low) level characteristics via the MANA ABM&S tool and the use of the ATLAS DST GUI provides further justification that movement away from the top down approach should be pursued and that the use of information extracted from the operational level results in a better forecast when determining logistical needs.

This hypothesis is primarily dependent on the intangible assessment of what is enough in terms of data that can adequately provide answers to the questions or requirements that face a DM. The author believes the military as a whole needs to have some control of how it approaches and makes decisions. The desire by the military to be flexible and the need to be dynamic to meet varying operational environments supports this notion. Because the future is uncertain and situations are constantly in flux, uncomplicated, suitable, simple, and easily adaptable models should be used to provide a holistic picture that can be easily filtered and managed at the user level. The ATLAS method seeks to do this and has shown this with the capabilities of the ATLAS DST GUI as a decision support tool.

The third hypothesis incorporates a concept that was first introduced in 1965, but has recently gained significant interest in the research community as a way to incorporate the HITL or at least mimic human decision making processes and is stated below.

> *Fuzzy Logic principles ought to be used to account for human in the loop, i.e. the decision maker's, cognitive processes when utilizing a DSS to determine a course of action.*

> Contribution: Incorporating this technique requires only a small pool of users, not all of them to agree upon rules and membership functions used to determine fitness or viability of a presented case set. As long as the fuzzy inference system is maintained to account for operational environment changes, applying this technique significantly reduces the work load placed on the logistician and allows him/her to reduce their decision making process time; especially at the operational level.

Advances in computing have made this ability somewhat a reality, but there exists no "right" way to incorporate *fuzzy* principles. The author's approach provides some flexibility that can provide an alternative view on information that may have otherwise been discounted as relevant to the problem. This approach again seeks to engage the HITL, because although useful, fuzzy logic and other techniques that attempt to model a human's cognitive abilities have limitations and have only been able to incorporate basic human processes.

The final and fourth hypothesis below sought to try and prove that the use of statistical methods is beneficial in shedding some information in a complex and sometimes information void or saturated environment.

Optimization techniques should be used to account for uncertainty in information and aid in providing transparency and justification for a COA taken.

Contribution: Further provides support for the importance of applying statistical analysis and visualization techniques that can be used to provide both the operational and top level planners a level of detail not readily seen. Displays that the desire for rapid design space exploration by military leaders is achievable and at the lower level via the ATLAS DST GUI and its supporting database structure.

While optimization is not the intent, finding a flexible and robust solution can be accomplished by instituting optimization techniques to provide information about the environment. This hypothesis is difficult to prove because only time will tell if these techniques are applicable for this environment. The impacts and insights from this works analysis look promising and has advanced insight in an arena where understanding the impacts on the outcomes based on specific logistical decisions at the small unit level are murky at best. This area of research will continue to be a source of work for a long time simply because it is a complex, dynamic, and an evolving environment that is affected by a variety of factors that are almost impossible to holistically account for.

6.2 When Should the ATLAS Method and Tools be Used

The ATLAS method should be used in the pre-planning stages of small unit operations by a logistics DM to try and forecast the unit's operational and logistical requirements. It potentially could be used to aggregate the need over the realm of several small units performing distributed operations or combat like operations similar to the scenario used in this work. This currently is the only scenario type available in the ATLAS DST and is also only applicable to the ranges established in Table 12. Consequently, the NN surrogate is only valid within these ranges and should not be utilized outside of them. The ATLAS DST GUI prevents the user from initiating queries outside these bounds, but the database and tool could be expanded by the addition of cases and additional scenario modeling to represent the “new” environment. The ATLAS method itself should be applicable to any variety of environments of complex and nonlinear effects and where historical or case based data can be utilized to enable a decision support system framework. Slight modifications of the method may be necessary depending on the environment, characteristics, and desired results by the end user.

6.3 Recommendations and Future Work

Forecasting and making predictions are one in the same, they are literally impossible to get accurate 100 % of the time. This is true even with stable and known demand. According to Dr. Gary Horne at the Naval Post Graduate School: “You can’t really predict anything, but if you look at enough possibilities, you can begin to understand” [66]. In Ilachinski’s book, Artificial War, Prussian General Carl von Clausewitz states that “absolute, so-called mathematical, factors never find a firm basis in military calculations” and highlights the uncertain and complex operational landscape facing the military today [54]. Yet in all our analysis and approaches today for solving problems, we seek to use statistics and mathematical modeling to describe outcomes, and then apply numerical techniques to predict and forecast future events. What is the solution to this problem of logistics determination from the small unit level to the operational level for the military? Is it solvable? Human nature and desire commands us to understand and to try and solve this problem.

An approach was pursued in this work to bring information and knowledge to the decision maker in a cognizant and useable fashion in a quicker manner. Case based reasoning, fuzzy principles, neural networks, and visualization techniques were joined for this effort. This work seeks a new approach to a relatively new, yet old problem, for the military and its leader; adequate and sustainable logistics support that satisfies diverse operational requirements. Modeling and simulation, although not novel, continues to evolve as abilities to model real world effect appear. Efforts to create a more dynamic, realistic, and autonomous ABM should be pursued and would further aid exploring plausible scenarios and their outcomes based on certain decision making actions.

Continued research and incorporation of new or innovative ways to incorporate cognitive reasoning processes is important to gaining insight into the dynamic systems that encompass the military operational environment. Accounting for uncertainty and variability in information can always be improved to facilitate that the best information and host of solutions is available to the DM for review and selection of feasible and viable courses of action. Other areas the author would recommend attention to consist of increasing the fidelity of the M&S environment, expansion beyond the operational scenario to encompass other scenario environments, and incorporation of real world scenario data to help create a true operational database.

There likely will never result in a catch all solution to forecasting logistical needs and requirements for the military with complete accuracy, but incremental improvements are useful non the less. These improvements should be beneficial in conserving resources, increasing mission success, and potentially saving soldiers lives as the military transforms itself to meet new and emerging threats at home and abroad.

Appendix A
SOFTWARE SELECTION TOOL

A 1. Introduction

The purpose of this appendix is to provide detailed documentation of a developed software selection tool (SST) utilized to determine specific software or software combinations specifically for use in this thesis and centered on the ability to aid in the predicting or forecasting for a network centric system of logistics. Software research was conducted to assess a viable software tool from a pool of various software suites. This appendix will provide a description of the developed tool, instructions on usage and calculations, and finally the results/selection of a software package to model the logistics environment. After comparing eight software applications within the developed tool (Any Logic, Arena, MatLab/Simulink, Vensim, EXTEND, Pythagoras, Map Aware Non-Uniform Automate (MANA) and EINSTein), and performing quantitative comparative assessments on software, metrics, and capabilities as well as a sensitivity analysis, the final selection identified MANA the most desirable software to utilize given the problem description and requirements.

A 2. Software Research

Three *types/levels* of modeling and simulation environments as well as results visualization were identified prior to determining which specific software to compare against customer requirements within the developed SST: (1) Stand alone scripting languages in conjunction with stand alone visualization tools; (2) Scripting languages that include some visualization capabilities; (3) Full package scripting, customization and visualization. Each of these types will be discussed in further detail below.

The first type of modeling and simulation environment with visualization of results requires the union of a scripting language – to provide the modeling and simulation environment – with a visualization suite to illustrate results. Examples of basic scripting languages are C, C++, FORTRAN, ADA, Pascal; and more advanced languages are SimPy and Vensim; whereas visualization tool examples are VisSim, SimProcess and SimCreator. A short description of the more advanced languages and visualization tools follow below. The information on each software option was derived from the software's website:

- SimPy (Simulation in Python) is an object-oriented, process-based discrete-event simulation language based on standard Python. It provides components of a simulation model including processes for active components; as well as processes for passive components that form limited capacity congestion points. SimPy also provides monitor variables to aid in gathering statistics and random variants are provided by the standard Python random module. SimPy is easily interfaced with other packages, such as plotting, statistics, GUI, spreadsheets and databases.

- Vensim is a system simulator from Ventana Systems, based on the world dynamics approach. Vensim is an integrated framework for conceptualizing, building, simulating, analyzing, optimizing and deploying models of dynamic systems. It uses a workbench-toolbox design that combines the simplicity of visual models with easy access to a host of powerful model simulation and analysis tools. This yields an increase in the speed of work and the quality of results.

- VisSim is a visual block diagram language for nonlinear dynamic simulation. VisSim allows users to create their own blocks in C/C++, FORTRAN, ADA, or Pascal with additional tools allowing for real-time analog and digital I/O for real-time simulation, embedded C code generation, optimization, neural nets, scaled fixed point and IIR and FIR filter design.
- Simprocess is an object-oriented, process modeling and analysis tool. It combines the simplicity of flowcharting with the power of simulation, statistical analysis, Activity-Based Costing (ABC) and animation.
- SimCreator is a graphical simulation and modeling system. Its main focus is the simulation of continuous time systems with an interface similar to MathWorks' Simulink. SimCreator facilitates complex models through connection of simpler models and nesting. This graphical specification is then translated in C code.

The next level of environment and visualization includes scripting languages with some existing visualization capabilities. Examples of such are MatLab with SimuLink and Mathematica. Each of these is described in further detail below:

- MatLab is the leading language for technical computing, DSP and control design among others. SIMULINK provides a graphical interface to some MatLab functions enabling the user to graphically design models and control systems.

Finally, the upper level tools identified are full package environment, customization, and visualization. Examples of these include Arena, EXTEND, MANA, Pythagoras and EINSTein. The last three were discussed in the body of this thesis in Section 2.14.

- Arena Professional simulation software is most effective when analyzing complex, medium to large-scale projects involving highly sensitive changes related to supply chain, manufacturing, processes, logistics, distribution, warehousing and service systems. Additionally, Arena PE is used to create customized simulation modeling products. These templates focus on specific applications or industries and consist of "libraries" of modeling objects enabling significantly simpler and faster development of models requiring repeat logic.
- EXTEND allows the creation of dynamic models from building blocks, exploration of the processes involved, and observation of relationships. Blocks are the basic model-building components in Extend with each block representing a modeled process containing unique procedural information. Blocks are grouped into libraries according to function. Extend allows the development of customized components and has an interactive and graphical architecture that is combined with a robust development environment.

Using the information obtained above about specific software packages, examples from each level of software type are used in the SST, totaling eight software packages to compare: Any Logic, Arena, MatLab/Simulink, Vensim, EXTEND, Pythagoras, MANA, and EINSTein. Each of the eight software options is comparatively analyzed with regard to specific requirements/needs.

A 3. Software Selection Tool

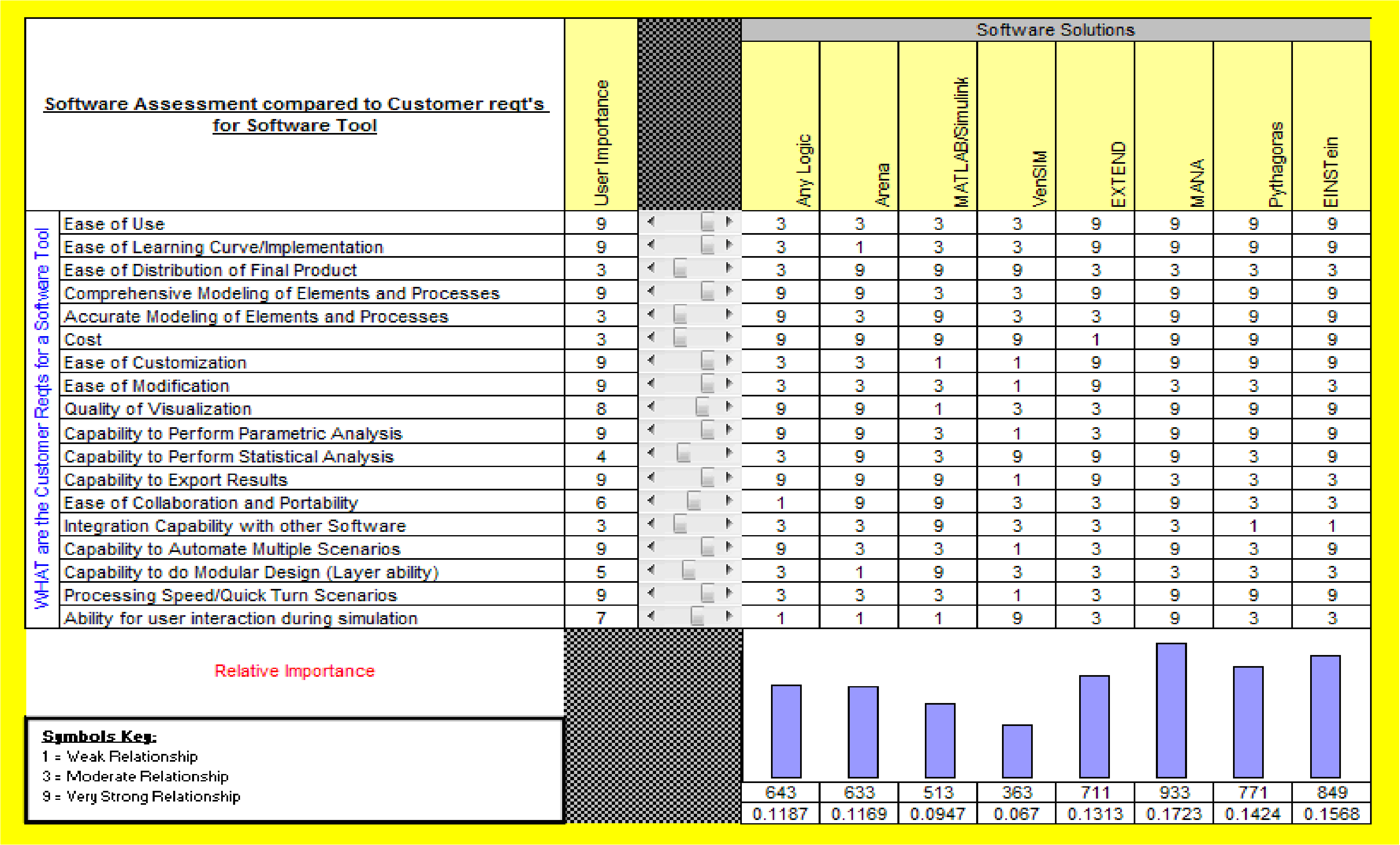
Sheet One: Software Assessment

Visual Orientation: Referring to the SST (sheet one), cells C4 through C21 contain the identified customer requirements, but are also presented below in Table 21 for ease of reference.

Table 21: Identified customer requirements

1	Ease of Use
2	Ease of Learning Curve/Implementation
3	Ease of Distribution of Final Product
4	Comprehensive Modeling of Elements and Processes
5	Accurate Modeling of Elements and Processes
6	Cost
7	Ease of Customization
8	Ease of Modification
9	Quality of Visualization
10	Capability to Perform Parametric Analysis
11	Capability to Perform Statistical Analysis
12	Capability to Export Results
13	Ease of Collaboration and Portability
14	Integration Capability with other Software
15	Capability to Automate Multiple Scenarios
16	Capability to do Modular Design (Layer ability)
17	Processing Speed/Quick Turn Scenarios
18	Ability for user interaction during simulation

Cells D4 through D22 present the user ranking (one-three-nine scale) for each of the aforementioned requirements. The importance rankings documented in cells S4 to S21 and T4 to T21. Cells G3 through N3 house each of the software options under evaluation. Each software option's ranking for each requirement is presented in cells G4 to N21 and correspond to each software option's column. Results for the rankings appear in cells G29 to N30, with a graphical representation appearing just above it. This is pictured below in Figure 68.



	Software Assessment compared to Customer reqt's for Software Tool	User Importance		Software Solutions							
				Any Logic	Arena	MATLAB/Simulink	VenSIM	EXTEND	MANA	Pythagoras	EINSTein
WHAT are the Customer Reqts for a Software Tool	Ease of Use	9		3	3	3	3	9	9	9	9
	Ease of Learning Curve/Implementation	9		3	1	3	3	9	9	9	9
	Ease of Distribution of Final Product	3		3	9	9	9	3	3	3	3
	Comprehensive Modeling of Elements and Processes	9		9	9	3	3	9	9	9	9
	Accurate Modeling of Elements and Processes	3		9	3	9	3	3	9	9	9
	Cost	3		9	9	9	9	1	9	9	9
	Ease of Customization	9		3	3	1	1	9	9	9	9
	Ease of Modification	9		3	3	3	1	9	3	3	3
	Quality of Visualization	8		9	9	1	3	3	9	9	9
	Capability to Perform Parametric Analysis	9		9	9	3	1	3	9	9	9
	Capability to Perform Statistical Analysis	4		3	9	3	9	9	9	3	9
	Capability to Export Results	9		9	9	9	1	9	3	3	3
	Ease of Collaboration and Portability	6		1	9	9	3	3	9	3	3
	Integration Capability with other Software	3		3	3	9	3	3	3	1	1
	Capability to Automate Multiple Scenarios	9		9	3	3	1	3	9	3	9
	Capability to do Modular Design (Layer ability)	5		3	1	9	3	3	3	3	3
	Processing Speed/Quick Turn Scenarios	9		3	3	3	1	3	9	9	9
	Ability for user interaction during simulation	7		1	1	1	9	3	9	3	3
	Relative Importance										
				643	633	513	363	711	933	771	849
				0.1187	0.1169	0.0947	0.067	0.1313	0.1723	0.1424	0.1568

Symbols Key:
1 = Weak Relationship
3 = Moderate Relationship
9 = Very Strong Relationship

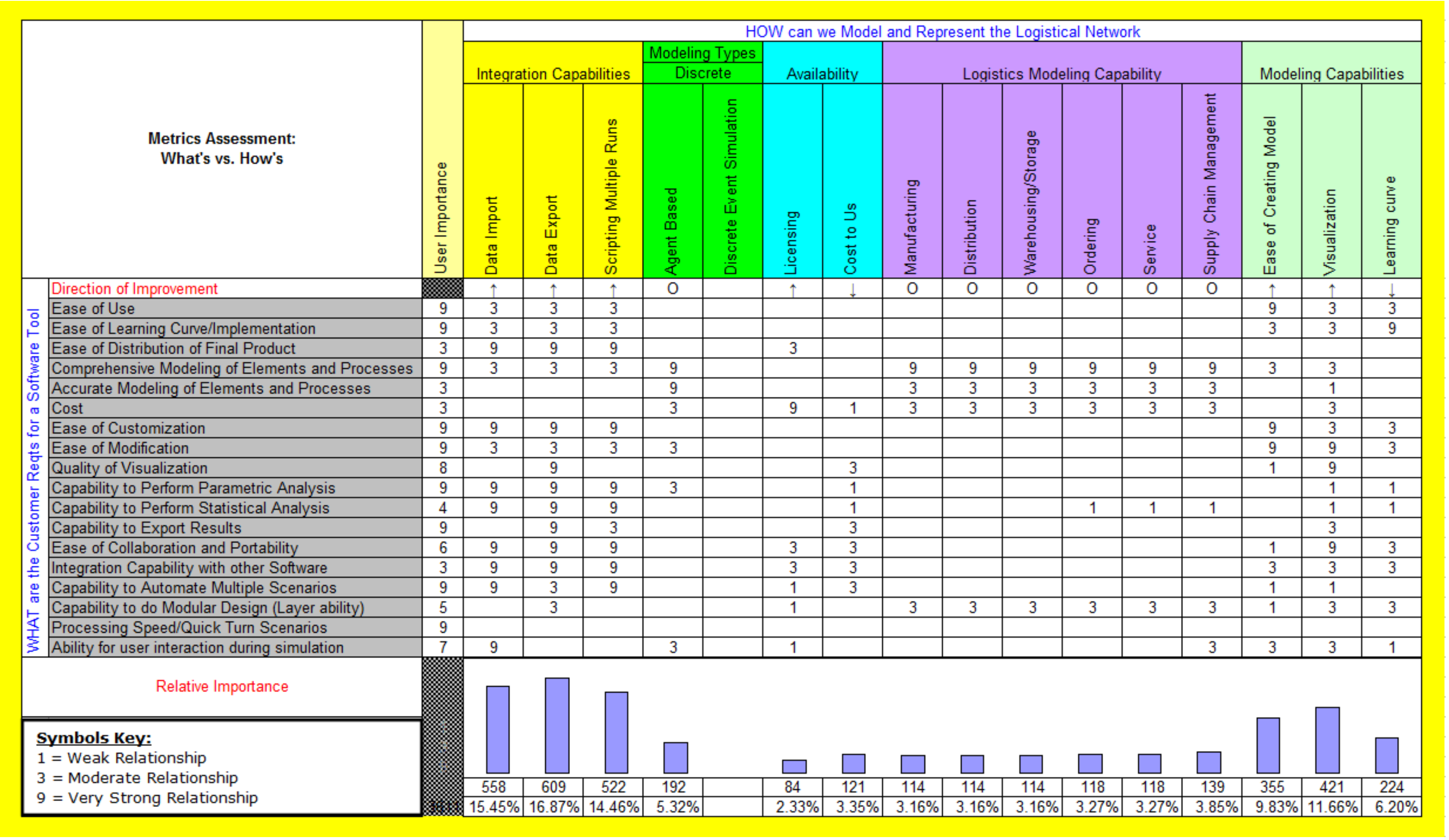
Figure 68: Software Assessment Sheet of SST

Description: The software assessment sheet compares each software tool against identified user requirements. User requirements have weightings based on overall importance and each software option is ranked against each user requirement. Both user requirement importance and software rankings are based on a one-three-nine scale, where one corresponds to a weak relationship, three a medium relationship, and nine a strong relationship.

Results: Results were obtained by multiplying the user importance of a particular requirement by the ranking the software received for that requirement, and then summing all values for each software option. A percentage version of this result was obtained by dividing the numerical result for each software option by the sum of the results for all software options. The software with the highest value is most optimal on the software assessment. For the software assessment using the user importance rankings, MANA ranks highest with numerical value of 933 and an overall result percentage of 17.22 % when compared to the other eight software applications.

Sheet Two: Metrics Assessment

Visual Orientation: The metrics assessment sheet (sheet two) presents the aforementioned user requirements in cells C7 through C24 with the same importance rankings in cells D7 to D24 (these rankings are automatically updated from the software assessment based upon user inputs). Options of how to represent the modeling environment are broken down into integration capabilities, modeling types, availability, and logistical modeling capability; these appear in cells F4 to W4. Under each of these options are specific metrics, for example, data import and export abilities fall under integration capabilities; and, licensing and cost fall under availability; these sublevel options are presented in cells F5 to U5 (totaling 16 options). The same ranking system is used as is in the software assessment sheet with the exception that some metrics show improvement in a decreasing trend (nine-three-one). This is identified in cells N6 and U6 for the metrics cost and learning curve; this is taken into account in calculating results. Each modeling option's ranking for each user requirement is presented in cells F7 to U24 and correspond to each modeling option's column. Results for the rankings appear in cells F31 to U32. The metrics assessment sheet is pictured below in Figure 69.



Metrics Assessment:
What's vs. How's

HOW can we Model and Represent the Logistical Network

WHAT are the Customer Reqts for a Software Tool

		Integration Capabilities			Modeling Types Discrete		Availability		Logistics Modeling Capability						Modeling Capabilities		
	User Importance	Data Import	Data Export	Scripting Multiple Runs	Agent Based	Discrete Event Simulation	Licensing	Cost to Us	Manufacturing	Distribution	Warehousing/Storage	Ordering	Service	Supply Chain Management	Ease of Creating Model	Visualization	Learning curve
Direction of Improvement		↑	↑	↑	O		↑	↓	O	O	O	O	O	O	↑	↑	↓
Ease of Use	9	3	3	3											9	3	3
Ease of Learning Curve/Implementation	9	3	3	3											3	3	9
Ease of Distribution of Final Product	3	9	9	9			3										
Comprehensive Modeling of Elements and Processes	9	3	3	3	9				9	9	9	9	9	9	3	3	
Accurate Modeling of Elements and Processes	3				9				3	3	3	3	3	3		1	
Cost	3				3		9	1	3	3	3	3	3	3		3	
Ease of Customization	9	9	9	9											9	3	3
Ease of Modification	9	3	3	3	3										9	9	3
Quality of Visualization	8		9					3							1	9	
Capability to Perform Parametric Analysis	9	9	9	9	3			1								1	1
Capability to Perform Statistical Analysis	4	9	9	9				1				1	1	1		1	1
Capability to Export Results	9		9	3				3								3	
Ease of Collaboration and Portability	6	9	9	9			3	3							1	9	3
Integration Capability with other Software	3	9	9	9			3	3							3	3	3
Capability to Automate Multiple Scenarios	9	9	3	9			1	3							1	1	
Capability to do Modular Design (Layer ability)	5		3				1		3	3	3	3	3	3	1	3	3
Processing Speed/Quick Turn Scenarios	9																
Ability for user interaction during simulation	7	9			3		1							3	3	3	1
Relative Importance		558	609	522	192		84	121	114	114	114	118	118	139	355	421	224
	3611	15.45%	16.87%	14.46%	5.32%		2.33%	3.35%	3.16%	3.16%	3.16%	3.27%	3.27%	3.85%	9.83%	11.66%	6.20%

Symbols Key:
1 = Weak Relationship
3 = Moderate Relationship
9 = Very Strong Relationship

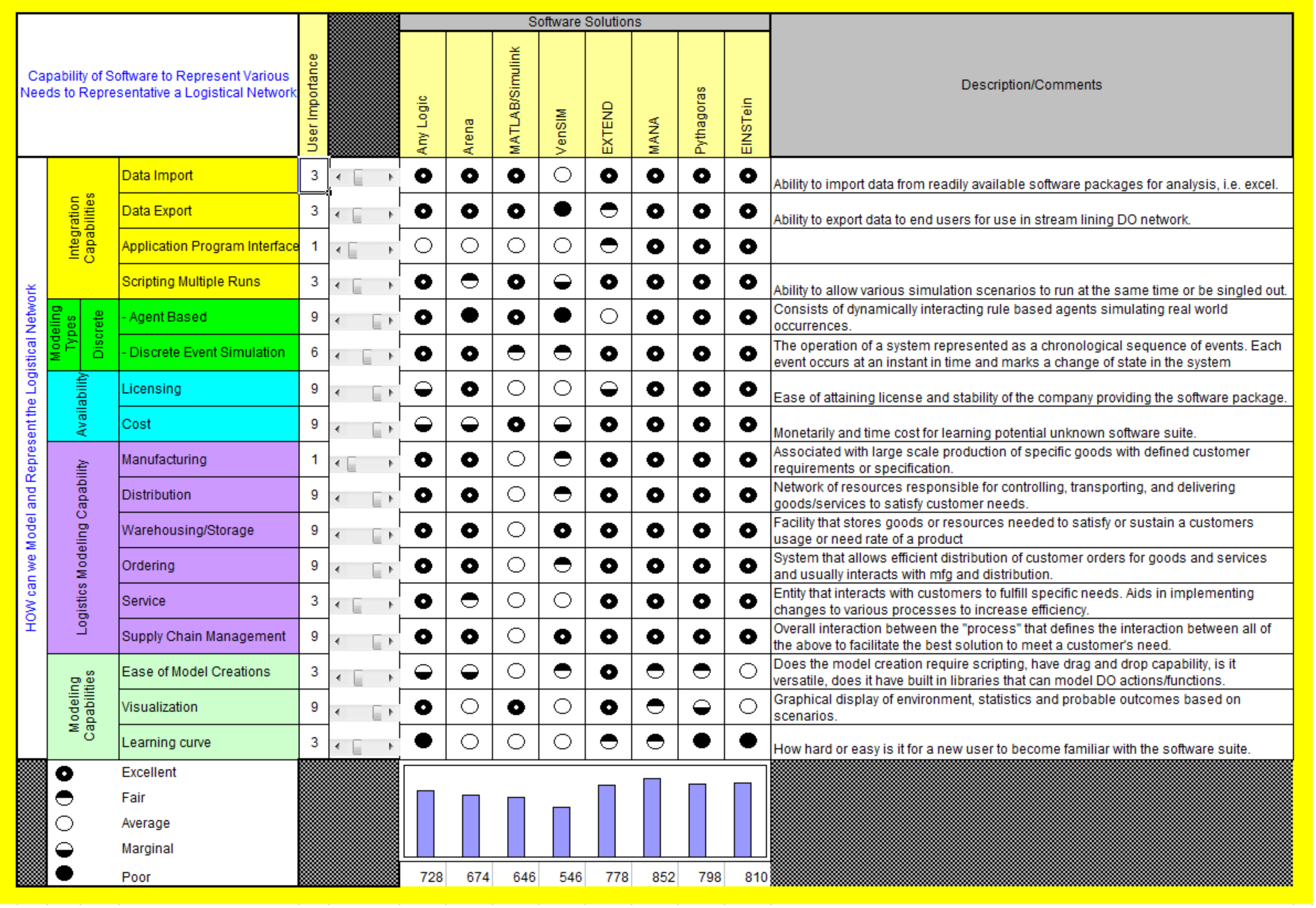
Figure 69: Metrics Assessment Sheet of SST

Description: The metrics assessment sheet (sheet two) identifies modeling characteristics that can be used to model and represent the logistical network. These modeling options are then compared and ranked against the identified user requirements. This sheet essentially determines how necessary a modeling characteristic is in order to fulfill each particular user requirement. Because not all modeling metrics/options have a relationship to each user requirement, some of the cells in this sheet remain blank.

Results: Results for this sheet were calculated in the same manner as in the software assessment sheet. The top three modeling options identified all fall under integration capabilities: data export ability (user ranking: 558 and 15.45%) data import ability (user ranking: 609 and 16.87%) and scripting multiple runs (user ranking: 522 and 14.46%).

Sheet Three: Capabilities Assessment

Visual Orientation: Each modeling option and its specific metrics appear in cells C6 through E20 with detailed descriptions of the modeling options appearing in cells Q6 through Q20. The user importance rankings for each of the 18 metrics appear in cells F6 through F20. Software options appear in cells I3 through P3. The rankings of each software option with regard to each user requirement appear in cells I4 through P20. Final numerical results are presented in cells I25 to P25. The capabilities assessment sheet is presented in Figure 70.

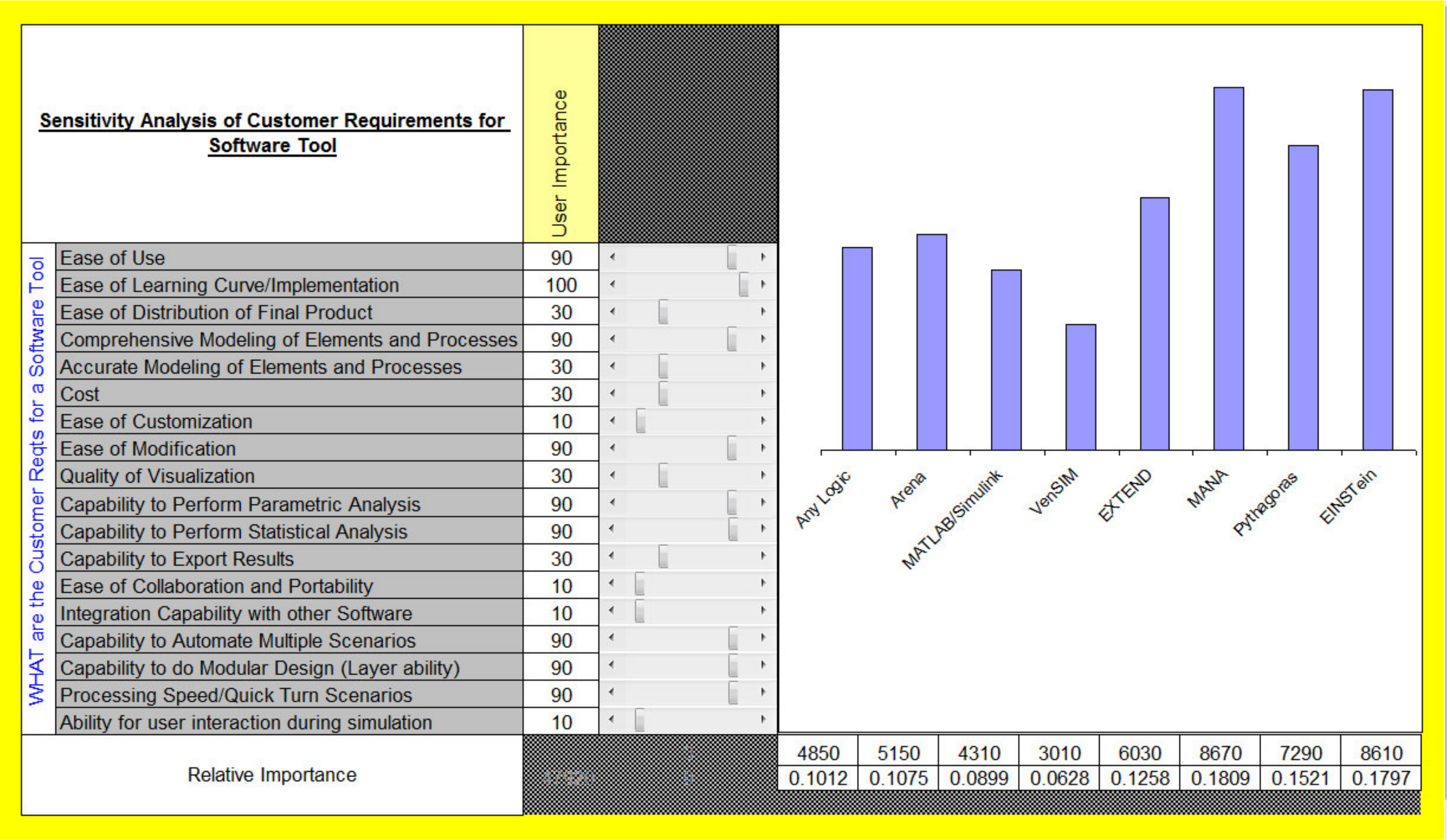


Description: The capabilities assessment sheet compares modeling characteristics of the metrics assessment sheet to each of the software options in a ranking style similar to the one used by *Consumer Reports Magazine*. That is, rather than the one-three-nine ranking system, a one-three-five-seven-nine system is utilized, where one is poor, three is marginal, five is average, seven is fair and nine is excellent. A user importance value is assigned to each of the modeling/metric options based on the one-three-nine system. Each software option is given a one-three-five-seven-nine weighting based upon its capability for each modeling option.

Results: The results for the capabilities assessment were calculated in the same manner as the previous sheets taking into account the new ranking system and omission of a percentage value for results. The top software option is MANA with a value of 852, followed by EINSTein (810), Pythagoras (798) and EXTEND (778).

Sheet Four: Sensitivity Analysis

Visual Orientation: Referring to the sensitivity analysis sheet, the previously noted user requirements are presented in cells C4 to C21 with associated user importance rankings in cells D4 to D21 and user ranking toggle bars in cells E4 through E21. User rankings for the user are located in cells S3 through T21. Numerical and percentage results are tabulated in cells G23 through N24 with a graphical representation of the results directly above. The sensitivity analysis sheet is presented below in Figure 71.



Sensitivity Analysis of Customer Requirements for Software Tool

WHAT are the Customer Reqts for a Software Tool

	User Importance
Ease of Use	90
Ease of Learning Curve/Implementation	100
Ease of Distribution of Final Product	30
Comprehensive Modeling of Elements and Processes	90
Accurate Modeling of Elements and Processes	30
Cost	30
Ease of Customization	10
Ease of Modification	90
Quality of Visualization	30
Capability to Perform Parametric Analysis	90
Capability to Perform Statistical Analysis	90
Capability to Export Results	30
Ease of Collaboration and Portability	10
Integration Capability with other Software	10
Capability to Automate Multiple Scenarios	90
Capability to do Modular Design (Layer ability)	90
Processing Speed/Quick Turn Scenarios	90
Ability for user interaction during simulation	10

	Any Logic	Arena	MATLAB/Simulink	VenSIM	EXTEND	MANA	Pythagoras	EINSTein
Relative Importance	4850	5150	4310	3010	6030	8670	7290	8610
	0.1012	0.1075	0.0899	0.0628	0.1258	0.1809	0.1521	0.1797

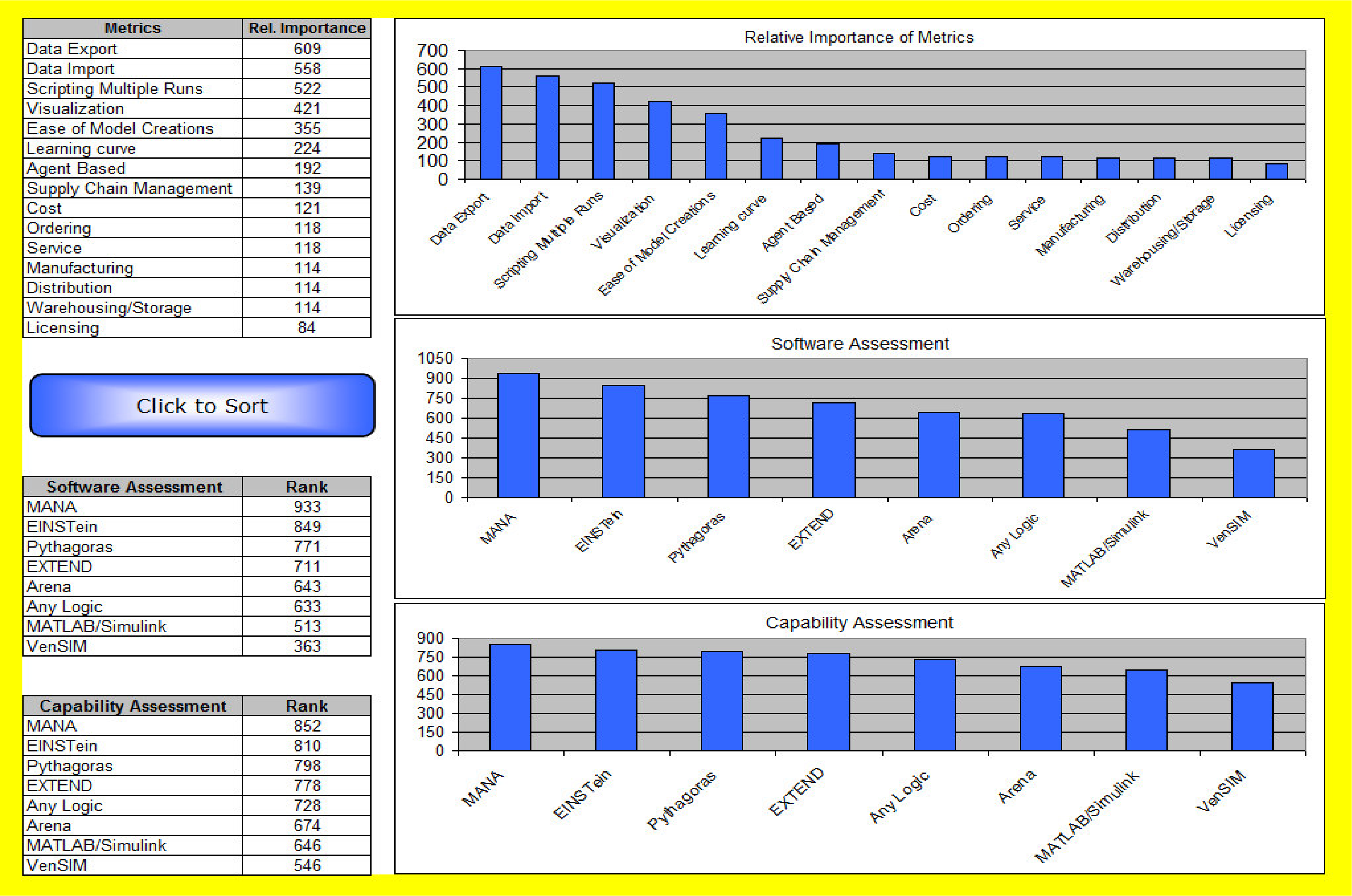
Figure 71: Sensitivity Analysis Sheet of SST

Description: The sensitivity analysis sheet looks very similar to the software assessment sheet with differences being the ranking system, toggle bars for ranking adjustment and an enlarged graphic of the software assessment. Instead of a one-three-nine ranking system, the values are on a hundred point scale: 10-30-90. This allows the user to determine how sensitive the software rankings are to small variations around a given value by toggling the provided bars one point at a time. A graphic is provided within the sensitivity analysis to illustrate how the software assessment changes.

Results: In addition to numerical and percentage results, a graphical result is presented in the sensitivity analysis sheet for visualization of sensitivity changes.

Sheet Five: Culmination of Results

Sheet five, shown in Figure 72, reiterates the results of the software, metrics and capabilities assessments with graphical representations for each.



Metrics	Rel. Importance
Data Export	609
Data Import	558
Scripting Multiple Runs	522
Visualization	421
Ease of Model Creations	355
Learning curve	224
Agent Based	192
Supply Chain Management	139
Cost	121
Ordering	118
Service	118
Manufacturing	114
Distribution	114
Warehousing/Storage	114
Licensing	84

Software Assessment	Rank
MANA	933
EINSTein	849
Pythagoras	771
EXTEND	711
Arena	643
Any Logic	633
MATLAB/Simulink	513
VenSIM	363

Capability Assessment	Rank
MANA	852
EINSTein	810
Pythagoras	798
EXTEND	778
Any Logic	728
Arena	674
MATLAB/Simulink	646
VenSIM	546

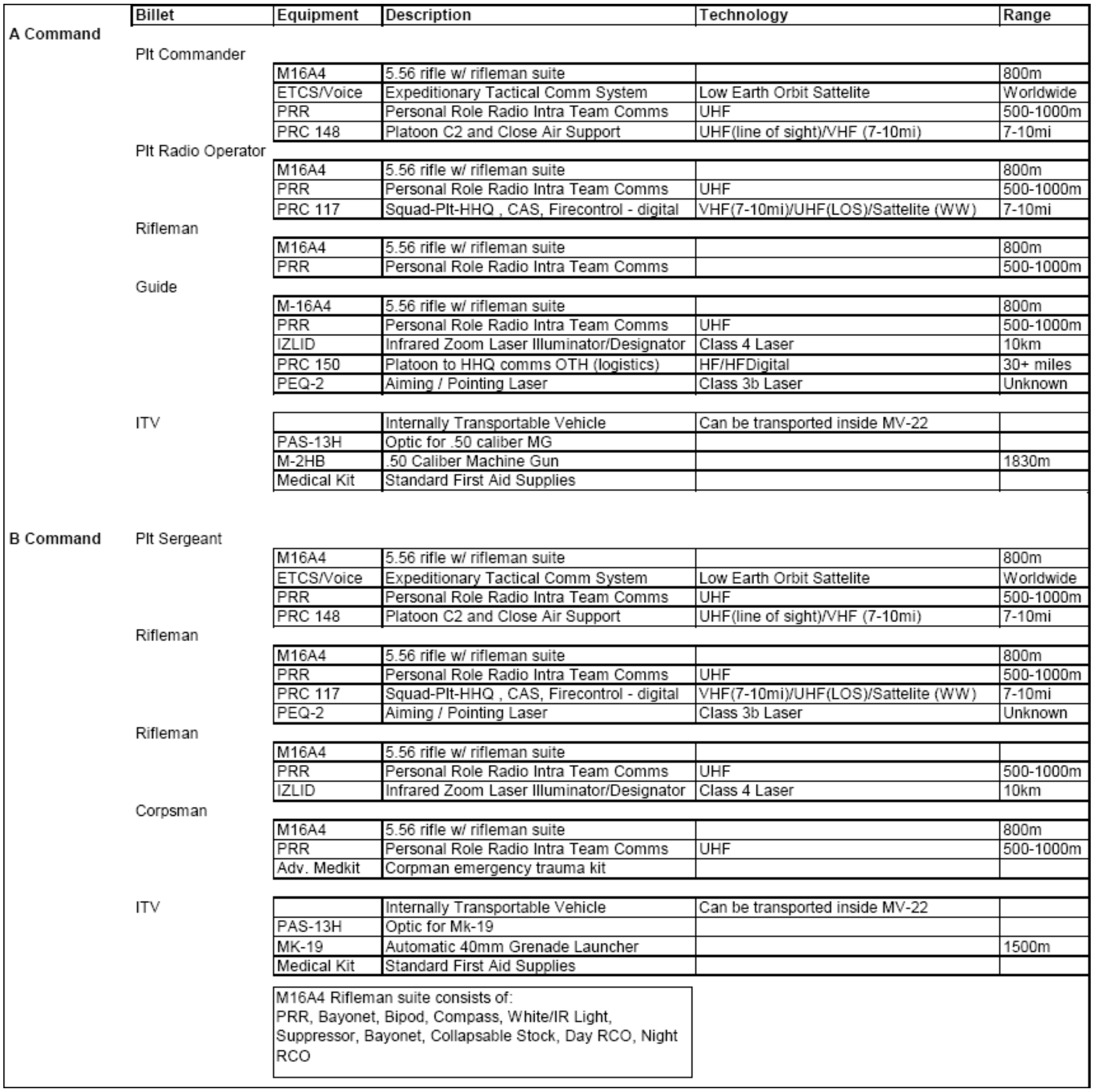
Figure 72: Final Results Sheet of SST

A 4. Conclusions

In conclusion, the assessment reveals that MANA is the top choice software application for modeling the environment. EINSTein, Pythagoras, and EXTEND, are very close to MANA and any of those applications could have been used if the scenario being investigated had not yet been created. In this case MANA already had a simulations scenario partially constructed for the author to modify for experimentation and analysis.

Appendix B
DO PLATOON CHARACTERISTICS & MANA MODEL DEVELOPMENT

B 1. DO Platoon Table of Organization and Equipment [8]



	Billet	Equipment	Description	Technology	Range
A Command					
	Plt Commander				
		M16A4	5.56 rifle w/ rifleman suite		800m
		ETCS/Voice	Expeditionary Tactical Comm System	Low Earth Orbit Sattelite	Worldwide
		PRR	Personal Role Radio Intra Team Comms	UHF	500-1000m
		PRC 148	Platoon C2 and Close Air Support	UHF(line of sight)/VHF (7-10mi)	7-10mi
	Plt Radio Operator				
		M16A4	5.56 rifle w/ rifleman suite		800m
		PRR	Personal Role Radio Intra Team Comms	UHF	500-1000m
		PRC 117	Squad-Plt-HHQ , CAS, Firecontrol - digital	VHF(7-10mi)/UHF(LOS)/Sattelite (WW)	7-10mi
	Rifleman				
		M16A4	5.56 rifle w/ rifleman suite		800m
		PRR	Personal Role Radio Intra Team Comms		500-1000m
	Guide				
		M-16A4	5.56 rifle w/ rifleman suite		800m
		PRR	Personal Role Radio Intra Team Comms	UHF	500-1000m
		IZLID	Infrared Zoom Laser Illuminator/Designator	Class 4 Laser	10km
		PRC 150	Platoon to HHQ comms OTH (logistics)	HF/HFDigital	30+ miles
		PEQ-2	Aiming / Pointing Laser	Class 3b Laser	Unknown
	ITV		Internally Transportable Vehicle	Can be transported inside MV-22	
		PAS-13H	Optic for .50 caliber MG		
		M-2HB	.50 Caliber Machine Gun		1830m
		Medical Kit	Standard First Aid Supplies		
B Command	Plt Sergeant				
		M16A4	5.56 rifle w/ rifleman suite		800m
		ETCS/Voice	Expeditionary Tactical Comm System	Low Earth Orbit Sattelite	Worldwide
		PRR	Personal Role Radio Intra Team Comms	UHF	500-1000m
		PRC 148	Platoon C2 and Close Air Support	UHF(line of sight)/VHF (7-10mi)	7-10mi
	Rifleman				
		M16A4	5.56 rifle w/ rifleman suite		800m
		PRR	Personal Role Radio Intra Team Comms	UHF	500-1000m
		PRC 117	Squad-Plt-HHQ , CAS, Firecontrol - digital	VHF(7-10mi)/UHF(LOS)/Sattelite (WW)	7-10mi
		PEQ-2	Aiming / Pointing Laser	Class 3b Laser	Unknown
	Rifleman				
		M16A4	5.56 rifle w/ rifleman suite		
		PRR	Personal Role Radio Intra Team Comms	UHF	500-1000m
		IZLID	Infrared Zoom Laser Illuminator/Designator	Class 4 Laser	10km
	Corpsman				
		M16A4	5.56 rifle w/ rifleman suite		800m
		PRR	Personal Role Radio Intra Team Comms	UHF	500-1000m
		Adv. Medkit	Corpman emergency trauma kit		
	ITV		Internally Transportable Vehicle	Can be transported inside MV-22	
		PAS-13H	Optic for Mk-19		
		MK-19	Automatic 40mm Grenade Launcher		1500m
		Medical Kit	Standard First Aid Supplies		

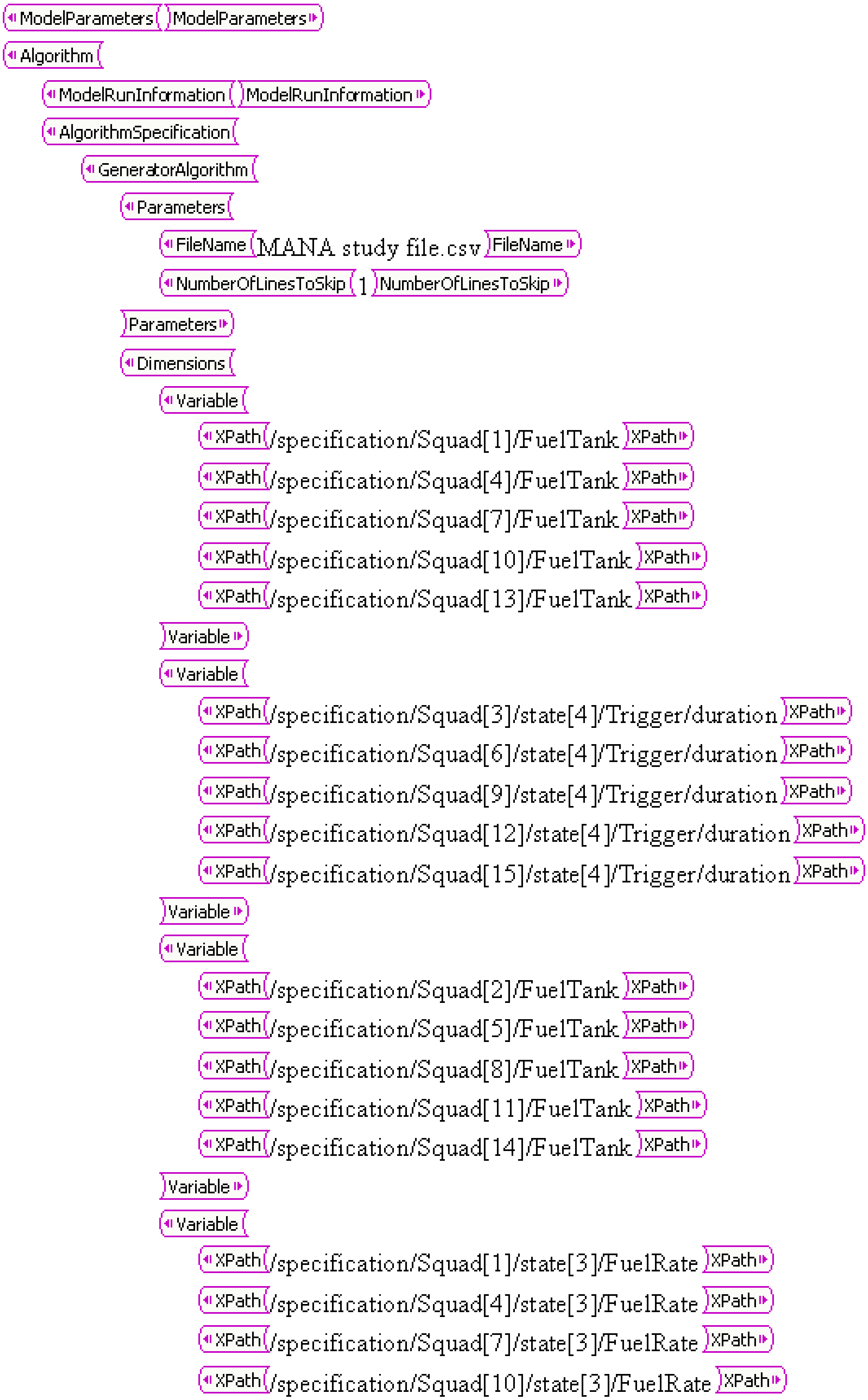
M16A4 Rifleman suite consists of:
PRR, Bayonet, Bipod, Compass, White/IR Light, Suppressor, Bayonet, Collapsable Stock, Day RCO, Night RCO

Squad Organization (3 per platoon)

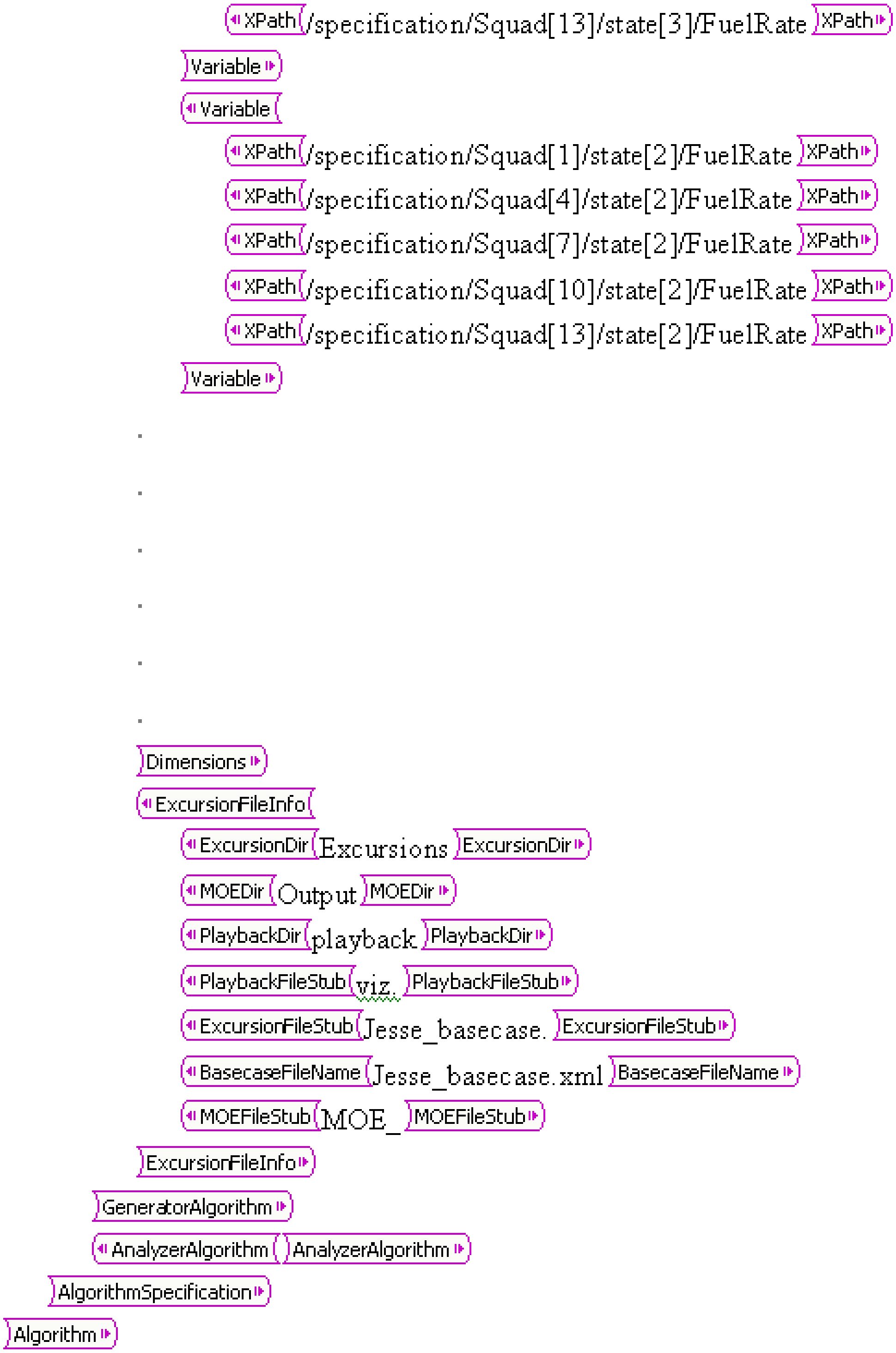
	Billet	Equipment	Description	Technology	Range
Squad C2					
	Squad Leader		SGT		
		M16 w/ M203	5.56 rifle w/ rifleman suite and 40mm grenade launcher		
		PRR	Personal Role Radio Intra Team Comms	UHF	500-1000m
		PRC 148	Platoon C2 and Close Air Support	UHF(line of sight)/VHF (7-10mi)	
		PEQ-2	Aiming / Pointing Laser	Class 3b Laser	Unknown
	Rifleman		PVT-LCPL		
		M16A4	5.56 rifle w/ rifleman suite		
		PRR	Personal Role Radio Intra Team Comms	UHF	500-1000m
		ETCS/Voice	Expeditionary Tactical Comm System	Low Earth Orbit Sattelite	Worldwide
	Automatic Rifleman				
		M249 SAW	5.56 Machine Gun w/ spare barrel		1000m
		PRR	Personal Role Radio Intra Team Comms	UHF	500-1000m
	Rifleman		PVT-LCPL		
		M16A4	5.56 rifle w/ rifleman suite		
		PRR	Personal Role Radio Intra Team Comms	UHF	500-1000m
		IZLID	Infrared Zoom Laser Illuminator/Designator	Class 4 Laser	10km
	ITV				
		M240G	7.62 Machine Gun		1800m
		PVS-17	Night Sight		
		PRC 117	Squad-Plt-HHQ , CAS, Firecontrol - digital	VHF(7-10mi)/UHF(LOS)/Sattelite (WW)	
		GLTD II	Ground Laser Target Designator	10x magnification	20km
		Medical Kit	Standard First Aid Supplies		
Team 1					
	Team Leader		CPL		
		M16 w/ M203	5.56 rifle w/ rifleman suite and 40mm grenade launcher		800m
		PRR	Personal Role Radio Intra Team Comms	UHF	500-1000m
		PRC 148	Platoon C2 and Close Air Support	UHF(line of sight)/VHF (7-10mi)	
		PEQ-2	Aiming / Pointing Laser	Class 3b Laser	Unknown
	Rifleman		PVT-LCPL		
		M16A4			
		PRR	Personal Role Radio Intra Team Comms	UHF	500-1000m
		IZLID	Infrared Zoom Laser Illuminator/Designator	Class 4 Laser	10km
	Automatic Rifleman		PVT-LCPL		
		M249 SAW	5.56 Machine Gun w/ spare barrel		
		PRR	Personal Role Radio Intra Team Comms	UHF	500-1000m
	Rifleman		PVT-LCPL		
		M16A4	5.56 rifle w/ rifleman suite		800m
		PRR	Personal Role Radio Intra Team Comms	UHF	500-1000m
	ITV				
		M-2HB	.50 Caliber Machine Gun		1830m
		PAS-13H	Optic for M-2		
Team 2					
	Team Leader		CPL		
		M16 w/ M203	5.56 rifle w/ rifleman suite and 40mm grenade launcher		800m
		PRR	Personal Role Radio Intra Team Comms	UHF	500-1000m
		PRC 148	Platoon C2 and Close Air Support	UHF(line of sight)/VHF (7-10mi)	
		PEQ-2	Aiming / Pointing Laser	Class 3b Laser	Unknown
	Rifleman		PVT-LCPL		
		M16A4	5.56 rifle w/ rifleman suite		
		PRR	Personal Role Radio Intra Team Comms	UHF	500-1000m
		IZLID	Infrared Zoom Laser Illuminator/Designator	Class 4 Laser	10km
	Automatic Rifleman		PVT-LCPL		
		M249 SAW	5.56 Machine Gun w/ spare barrel		1000m
		PRR	Personal Role Radio Intra Team Comms	UHF	500-1000m
	Rifleman		PVT-LCPL		
		M16A4	5.56 rifle w/ rifleman suite		800m
		PRR	Personal Role Radio Intra Team Comms	UHF	500-1000m
	ITV				
		MK-19	Automatic 40mm Grenade Launcher		1500m
		PAS-13H	Optic for Mk-19		

B 2. Partial Sample of XStudy File

```xml
<Study>
    <Version>5</Version>
    <ModelIdentification>
        <ModelName>Mana</ModelName>
        <ModelVersion>
            <Major>4</Major>
            <Minor>1</Minor>
        </ModelVersion>
    </ModelIdentification>
    <StudyIdentification>
        <Name>DO Platoon with Indiv Sensors and Helos_2008-08-11</Name>
        <Description></Description>
    </StudyIdentification>
    <UserIdentification>
        <UserName>Jesse Hester</UserName>
        <EmailAddress>jesse.hester@asdl.gatech.edu</EmailAddress>
        <PhoneNumber>XXX-XXX-XXXX</PhoneNumber>
        <UserID>newid</UserID>
    </UserIdentification>
    <SubmissionParameters>
        <OriginatingMachine>harvest.nps.edu</OriginatingMachine>
        <Platform>Condor</Platform>
        <Evaluator>
            <Parameters>
                <MakeRuns>true</MakeRuns>
            </Parameters>
        </Evaluator>
    </SubmissionParameters>
    <ModelParameters>
        <RandomGeneratorClass>Default</RandomGeneratorClass>
        <RandomGeneratorMethod>Default</RandomGeneratorMethod>
        <NumberReplicates>5</NumberReplicates>
        <InitialRandomSeed>4</InitialRandomSeed>
        <PlaybacksWanted>no</PlaybacksWanted>
```



```
<ModelParameters></ModelParameters>
<Algorithm>
    <ModelRunInformation></ModelRunInformation>
    <AlgorithmSpecification>
        <GeneratorAlgorithm>
            <Parameters>
                <FileName>MANA study file.csv</FileName>
                <NumberOfLinesToSkip>1</NumberOfLinesToSkip>
            </Parameters>
            <Dimensions>
                <Variable>
                    <XPath>/specification/Squad[1]/FuelTank</XPath>
                    <XPath>/specification/Squad[4]/FuelTank</XPath>
                    <XPath>/specification/Squad[7]/FuelTank</XPath>
                    <XPath>/specification/Squad[10]/FuelTank</XPath>
                    <XPath>/specification/Squad[13]/FuelTank</XPath>
                </Variable>
                <Variable>
                    <XPath>/specification/Squad[3]/state[4]/Trigger/duration</XPath>
                    <XPath>/specification/Squad[6]/state[4]/Trigger/duration</XPath>
                    <XPath>/specification/Squad[9]/state[4]/Trigger/duration</XPath>
                    <XPath>/specification/Squad[12]/state[4]/Trigger/duration</XPath>
                    <XPath>/specification/Squad[15]/state[4]/Trigger/duration</XPath>
                </Variable>
                <Variable>
                    <XPath>/specification/Squad[2]/FuelTank</XPath>
                    <XPath>/specification/Squad[5]/FuelTank</XPath>
                    <XPath>/specification/Squad[8]/FuelTank</XPath>
                    <XPath>/specification/Squad[11]/FuelTank</XPath>
                    <XPath>/specification/Squad[14]/FuelTank</XPath>
                </Variable>
                <Variable>
                    <XPath>/specification/Squad[1]/state[3]/FuelRate</XPath>
                    <XPath>/specification/Squad[4]/state[3]/FuelRate</XPath>
                    <XPath>/specification/Squad[7]/state[3]/FuelRate</XPath>
                    <XPath>/specification/Squad[10]/state[3]/FuelRate</XPath>
```



```
        <XPath>/specification/Squad[13]/state[3]/FuelRate</XPath>
      </Variable>
      <Variable>
        <XPath>/specification/Squad[1]/state[2]/FuelRate</XPath>
        <XPath>/specification/Squad[4]/state[2]/FuelRate</XPath>
        <XPath>/specification/Squad[7]/state[2]/FuelRate</XPath>
        <XPath>/specification/Squad[10]/state[2]/FuelRate</XPath>
        <XPath>/specification/Squad[13]/state[2]/FuelRate</XPath>
      </Variable>
    .
    .
    .
    .
    .
    .
    </Dimensions>
    <ExcursionFileInfo>
      <ExcursionDir>Excursions</ExcursionDir>
      <MOEDir>Output</MOEDir>
      <PlaybackDir>playback</PlaybackDir>
      <PlaybackFileStub>viz.</PlaybackFileStub>
      <ExcursionFileStub>Jesse_basecase.</ExcursionFileStub>
      <BasecaseFileName>Jesse_basecase.xml</BasecaseFileName>
      <MOEFileStub>MOE_</MOEFileStub>
    </ExcursionFileInfo>
  </GeneratorAlgorithm>
  <AnalyzerAlgorithm></AnalyzerAlgorithm>
 </AlgorithmSpecification>
</Algorithm>
```

B 3. MANA Base Model Characteristics

The variables of interest have been identified in Table 16. This section will go in detail what the units of measure in MANA are and what each variable represents.

X1: DO Squads Days of Supply

This variable is relatively self explanatory. It accounts for the amount of supplies each person in the DO Platoon deploys with upon departing for their mission. In MANA this variable is represented by the agent's fuel tank. This capability in MANA allows the user to define a starting value as well as a variety of usage rates. In this scenario a DO squad member has three setting of usage. The first is when they soldier is moving about in the operational area conducting surveillance and patrol activities; this is the default state. The second and third will be covered in X4 and X5. A single day's supply is represented by 2880 fuel units and whatever the agent leaves with is what the agent will be resupplied back up to when a resupply mission is executed. So if an agent has three days of supply, when a resupply asset arrives at the agent's location the agent reverts back to three days of supply on hand

X2: Scheduled Resupply Rate

This variable is the rate, in days, at which the platoon will be resupplied by aerial assets based on expected consumption of the three staples of food water, and ammunition. While this number is a prescribed number that could be used for initial planning purposes, it does not limit the ability to conduct resupply operations earlier based on rapid resupply requests initiated by the DO platoon (X6). This likely occurs because of intense contact or exposure to enemy agents. One day is again represented by 2880 time units in MANA

X3: Sense and Respond Lead Time

This variable dictates how long the resupply point takes to process and execute a resupply mission once a request is received or the specified duration indicated in X2 is reached. It is a constant duration that is not modifiable in MANA once a simulation is executed.

X4: Consumption Rate: Enemy Contact

This variable control the amount of fuel the DO agent uses when enemy agents are within his situational awareness map. The agents situational awareness, whether through inter-agent communication reports, visual, or otherwise is when the agent is aware that enemy combatants are near by. This environment causes the agent to be on heightened alert and utilize slightly more fuel than when he was on patrol and surveillance activities.

X5: Consumption Rate: Shot at

This variable dictates the highest usage of fuel and is in effect when the agent is in direct enemy contact and is conducting combat operations. Even after completing these operations the agent will remain in this mode for a set amount of time to represent the environment of heightened alert by soldiers after combat.

X6: Rapid Request Setup Time

This is the amount of time in minutes that will take for the service support unit providing resupply to the DO platoon to react to a rapid resupply request.

X7: Time to conduct Resupply Mission

This variable represents the amount of time in minutes it takes for the aerial asset to fly from is support base, approximately 50 nautical miles away, to the DO platoon's location and resupply the requesting party. Each of the three squads and the two command elements have their own dedicated resupply assets to allow for data extraction in how many flights occur to each entity.

X8: Resupply Speed

This variable indicates how quickly the aerial resupply asset gets to the DO platoon to execute the resupply mission from its logistical supply base approximately 50 nautical miles away.

X9: Resupply Stealth

This variable represents how invisible the resupply assets are to enemy combatants.

X10: Friendly Inorganic Sensor Persistence

This variable represents how long a friendly force will keep an enemy agent on its situational awareness map via its sensors once it has been reported through the communications network

X11: Enemy Sensor Range

This is how far the enemy's sensor range is for detecting DO platoon agents along their path to infiltrate the border. The level of this ability dictates how well or how bad the enemy avoids DO agents while navigating the terrain in the model.

X12: Enemy Squad Size

This variable represents the number of enemy agents traveling together in an effort to reach their objective across the border. More agents tend to overwhelm the DO agents when attempting to eliminate the enemy combatants and allow passage of enemy agents

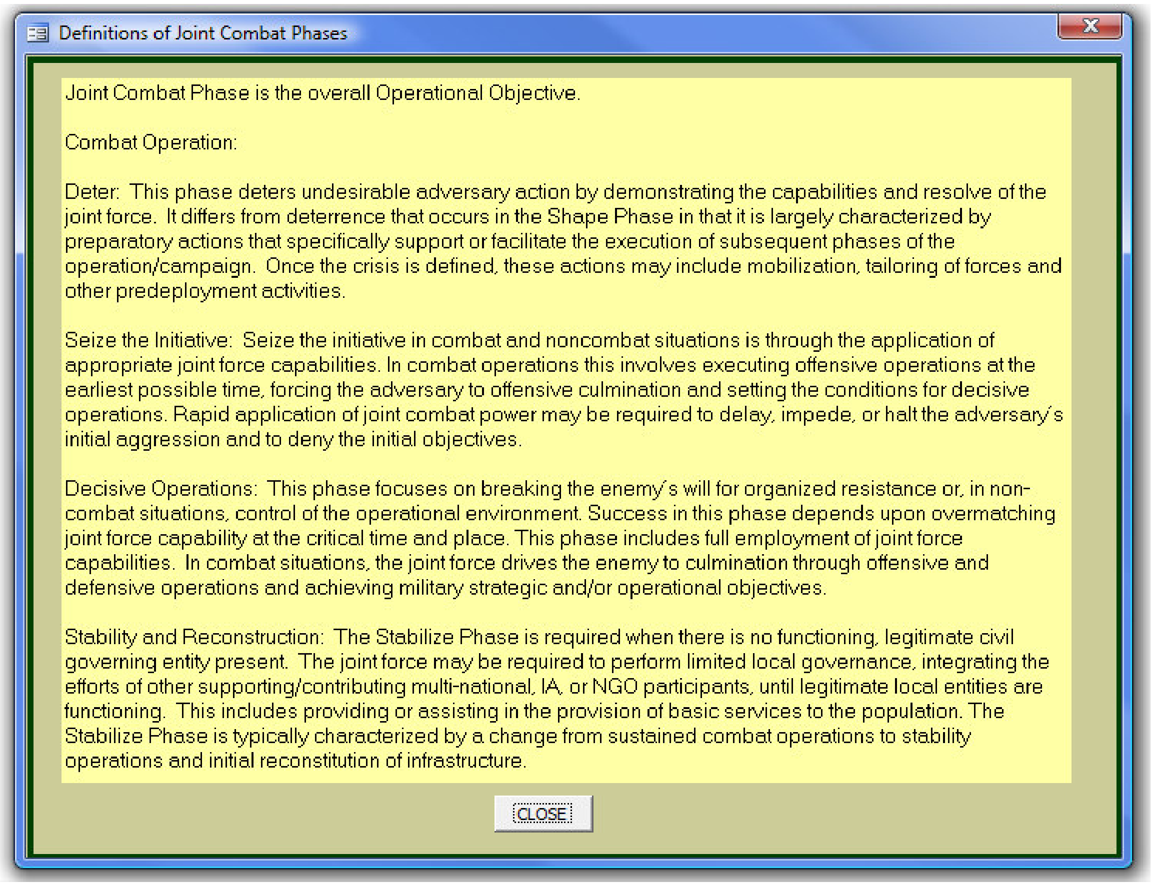
X13: Contact Persistence Enemy

This variable represent the amount of time an enemy agent will keep a DO agent on their situational awareness map and thus tend to avoid that agent if they are in their potential path to the objective across the border.

X14: Enemy Hits to Kill

This variable represents the number of direct hits required to kill and enemy agent, the higher the number the more well equipped and well trained the agent is at engaging the DO platoon agent and preventing themselves from being killed.

B 4. OPLOGPLN Additional Information



Joint Combat Phase is the overall Operational Objective.

Combat Operation:

Deter: This phase deters undesirable adversary action by demonstrating the capabilities and resolve of the joint force. It differs from deterrence that occurs in the Shape Phase in that it is largely characterized by preparatory actions that specifically support or facilitate the execution of subsequent phases of the operation/campaign. Once the crisis is defined, these actions may include mobilization, tailoring of forces and other predeployment activities.

Seize the Initiative: Seize the initiative in combat and noncombat situations is through the application of appropriate joint force capabilities. In combat operations this involves executing offensive operations at the earliest possible time, forcing the adversary to offensive culmination and setting the conditions for decisive operations. Rapid application of joint combat power may be required to delay, impede, or halt the adversary's initial aggression and to deny the initial objectives.

Decisive Operations: This phase focuses on breaking the enemy's will for organized resistance or, in non-combat situations, control of the operational environment. Success in this phase depends upon overmatching joint force capability at the critical time and place. This phase includes full employment of joint force capabilities. In combat situations, the joint force drives the enemy to culmination through offensive and defensive operations and achieving military strategic and/or operational objectives.

Stability and Reconstruction: The Stabilize Phase is required when there is no functioning, legitimate civil governing entity present. The joint force may be required to perform limited local governance, integrating the efforts of other supporting/contributing multi-national, IA, or NGO participants, until legitimate local entities are functioning. This includes providing or assisting in the provision of basic services to the population. The Stabilize Phase is typically characterized by a change from sustained combat operations to stability operations and initial reconstitution of infrastructure.

Appendix C
MATLAB CODE

C 1. MatLab Code for Post Processing of MANA Results

This appendix supplies several MatLab codes that were created by the author as well as by MatLab's GUI creator. The codes generated by the author were primarily used to read in and parse the data files generated by MANA that contained information about the operational landscape. The need for a parsing code was the result of the large output files that were generated by MANA, especially the agent position files. These files recorded several aspects of the simulation for every time step. Limitations in the amount of visible rows of EXCEL (65,000), the output file format, also necessitated a venue to read in the data and extract only what was pertinent for this research work.

The following MatLab code is used determine when a helicopter resupply occurred for any of the 5 entities in the MANA Scenario model. This is accomplished by taking the files, which are named with a numbering convention from case zero to the number specified. In the authors work 1000 cases were performed and each case was executed five times to generate some repeatability and average over the initial conditions specified by the DOE. Once the data file was opened by the code an algorithm that performed a matching of certain (X,Y) or position coordinates associated with a particular agent is performed. That information is stored in a matrix and when the entire output file from MANA is parsed that matrix is written out to a directory as another file with a similar naming convention for use by the ATLAS Tool GUI as the knowledge or database file that is needed for the ATLAS method.

```
for x=0:999;
    for z=1:5;
        % Open File at various locations
        f1='E:\oldmcdata\Studies\DO_Platoon\Output\MOE_-0-';
        f2=x;
        f2_=num2str(f2);
        f3='_pos_';
        f4=z;
        f4_=num2str(f4);
        f5='.csv';
        file = strcat(f1,f2_,f3,f4_,f5);
        C = textread(file,'','delimiter',',','commentstyle',
        'shell','headerlines',6);

        %position 1
        a=[C(:,2)==4 C(:,4)==104 C(:,5)==34];
        a1=sum(a,2);
        a2=floor(a1/3);
        clear a1

        %position 2
        b=[C(:,2)==9 C(:,4)==108 C(:,5)==102];
        b1=sum(b,2);
        b2=floor(b1/3);
        clear b1

        %position 3
        c=[C(:,2)==14 C(:,4)==105 C(:,5)==168];
        c1=sum(c,2);
        c2=floor(c1/3);
        clear c1

        %position 4
        d=[C(:,2)==17 C(:,4)==125 C(:,5)==118];
        d1=sum(d,2);
        d2=floor(d1/3);
        clear d1

        %position 5
        e=[C(:,2)==20 C(:,4)==133 C(:,5)==51];
        e1=sum(e,2);
        e2=floor(e1/3);
        clear e1

        big=[C a2 b2 c2 d2 e2];
        y=sortrows(big,[-6 -7 -8 -9 -10]);
        good=sum(sum([a2 b2 c2 d2 e2]));

        ynew=sortrows(y(1:good,1:end-5));

        f0='E:\oldmcdata\Studies\DO_Platoon\Output\Position\MOE_-0-';
        f6='_new';
        filenew=strcat(f0,f2_,f3,f4_,f6,f5);
        dlmwrite(filenew,ynew,'roffset',1);
```

```
        delete (file);
        clear a* b* c* d* e* C big y*;
    end;
end;
```

The following code is used to open the above newly created files to determine the amount of helicopter resupply flights that occurred during the simulation and identify at what time (in days) these flights occurred as well as to which agent the helicopter resupply flight was for in the simulations performed.

```
for x=0:999;
    for z=1:5;
        f1='C:\Documents and Settings\gtg730n\My
            Documents\MATLAB\Resupply Files 2\MOE_-0-';
        f2=x;
        f2_=num2str(f2);
        f3='_pos_';
        f4=z;
        f4_=num2str(f4);
        f5='_new.csv';
        file = strcat(f1,f2_,f3,f4_,f5);

        C = textread(file,'','delimiter',',','commentstyle',
        'shell','headerlines',1);

        dim = size(C,1);

        Resupply = (C(:,1)/2880);

        for j=1:dim
            if C(j,2)==4
                Resupply(j,2)=1;
            elseif C(j,2)==9
                Resupply(j,2)=2;
            elseif C(j,2)==14
                Resupply(j,2)=3;
            end
        end
        f0='C:\Documents and Settings\gtg730n\My Documents\MATLAB\Helo
        Sched\Helo_-0-';
        f6='.csv';
        filenew=strcat(f0,f2_,f3,f4_,f6);
        dlmwrite(filenew,Resupply);
    end;
end;
```

The code below is used to calculate the average number of total flights for each of the 5 iterations performed for a set of initial conditions as outline by the DOE for the Distributed Operations scenario.

```
for x=0:999;
    for z=1:5;
        % Open File at various locations
        f1='T:\MANA\Resupply Files\MOE_-0-';
        f2=x;
        f2_=num2str(f2);
        f3='_pos_';
        f4=z;
        f4_=num2str(f4);
        f5='_new.csv';
        file = strcat(f1,f2_,f3,f4_,f5);

        C = textread(file,'','delimiter',',','commentstyle',
        'shell','headerlines',1);

        a=sum(C(:,3)==2);
        A(z,1)=a;
        num_flights = round(sum(A)/5);

    end;

    output=num_flights;
    dlmwrite('C:\MANA Helo flights.csv',output,'-append');

end;
```

C 2. GUI Code

The bulk of this code is automatically generated by MatLab as the ATLAS Tool GUI is created through the user interface when adding components, whether this is static text, drop down boxes, pictures, or a variety of execution buttons. The author did modify some of this code to keep user inputs within the realm of the ranges set for each on the input variables, specified in Table 12, and to keep confidence levels between the range of zero and 100 %. This coding can be seen in the `function X1_Callback` or `function`

`C1_Callback` as examples. This code is also modified to call on another code named ***guiengine.m*** that contains the bulk of the algorithms used to execute accessing the database to conduct the case based reasoning application and evaluation, apply the fuzzy logic processes discussed in the body of this work, utilize the artificial neural net created from the input output data, and perform a variety of visualization capabilities available to the ATLAS Tool user.

```
function varargout = DST_ATLAS_Helo_Pred(varargin)
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
DST_ATLAS_Helo_Pred

% Last Modified by GUIDE v2.5 19-Nov-2008 11:51:06

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @DST_ATLAS_Helo_Pred_OpeningFcn,
...
                   'gui_OutputFcn',  @DST_ATLAS_Helo_Pred_OutputFcn,
...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before DST_ATLAS_Helo_Pred is made visible.
function DST_ATLAS_Helo_Pred_OpeningFcn(hObject, eventdata, handles,
varargin)

% ---- Calls to guiengine.m ----
[handles] = guiengine(handles,'initialize');

% Choose default command line output for DST_ATLAS_Helo_Pred
handles.output = hObject;
```

```
% Update handles structure
set(hObject,'toolbar','figure');
guidata(hObject, handles);


% --- Outputs from this function are returned to the command line.
function varargout = DST_ATLAS_Helo_Pred_OutputFcn(hObject, eventdata,
handles)
varargout{1} = handles.output;
% ---------------------Populate Pictures in GUI ---------------------

% --- Executes during object creation, after setting all properties.
function Icon_CreateFcn(hObject, eventdata, handles)
% Hint: place code in OpeningFcn to populate Icon

% --- Executes during object creation, after setting all properties.
function Graph_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Graph (see GCBO)
% Hint: place code in OpeningFcn to populate Graph

% --- Executes during object creation, after setting all properties.
function Resupply_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Resupply (see GCBO)
% Hint: place code in OpeningFcn to populate Resupply
% -----------------------------------------------------------------

% READ IN X1 TO X14 VARIABLES FOR DETERMINING CASE CHARACTERISTICS

% --- X1 Functions ---
function X1_Callback(hObject, eventdata, handles)
X1 = str2double(get(hObject,'String'));

% Perform Check that X1 is with in Data Base Parameters
if isnan(X1)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if (X1 < 1 || X1 > 5);
    set(hObject,'String','')
    errordlg('Must be at between 1 and 5 days','Error');
end

% --- Executes during object creation, after setting all properties.
function X1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X1_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to X1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.X1, 'String', '');
```

```
% --- X2 Functions ---
function X2_Callback(hObject, eventdata, handles)

X2 = str2double(get(hObject,'String'));

if isnan(X2)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if (X2 < 1 || X2 > 5);
    set(hObject,'String','')
    errordlg('Must be at between 1 and 5 min','Error');
end

function X2_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X2_ButtonDownFcn(hObject, eventdata, handles)
set(handles.X2, 'String', '');

% --- X3 Functions ---
function X3_Callback(hObject, eventdata, handles)

X3 = str2double(get(hObject,'String'));

if isnan(X3)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if (X3 < 0 || X3 > 720);
    set(hObject,'String','')
    errordlg('Must be at between 0 and 720 min','Error');
end

function X3_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X3_ButtonDownFcn(hObject, eventdata, handles)
set(handles.X3, 'String', '');

% --- X4 Functions ---
function X4_Callback(hObject, eventdata, handles)

X4 = str2double(get(hObject,'String'));
```

```
if isnan(X4)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if (X4 < 1 || X4 > 50);
    set(hObject,'String','')
    errordlg('Must be at between 1 and 50 units/time','Error');
end

function X4_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X4_ButtonDownFcn(hObject, eventdata, handles)
set(handles.X4, 'String', '');

% --- X5 Functions ---
function X5_Callback(hObject, eventdata, handles)

X5 = str2double(get(hObject,'String'));

if isnan(X5)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if (X5 < 1 || X5 > 50);
    set(hObject,'String','')
    errordlg('Must be at between 1 and 50 units/time','Error');
end

function X5_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X5_ButtonDownFcn(hObject, eventdata, handles)
set(handles.X5, 'String', '');

% --- X6 Functions ---
function X6_Callback(hObject, eventdata, handles)

X6 = str2double(get(hObject,'String'));

if isnan(X6)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if (X6 < 1 || X6 > 360);
    set(hObject,'String','')
```

```
    errordlg('Must be at between 1 and 360 min','Error');
end

function X6_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X6_ButtonDownFcn(hObject, eventdata, handles)
set(handles.X6, 'String', '');

% --- X7 Functions ---
function X7_Callback(hObject, eventdata, handles)

X7 = str2double(get(hObject,'String'));

if isnan(X7)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if (X7 < 1 || X7 > 60);
    set(hObject,'String','')
    errordlg('Must be at between 1 and 60 min','Error');
end

function X7_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X7_ButtonDownFcn(hObject, eventdata, handles)
set(handles.X7, 'String', '');

% --- X8 Functions ---
function X8_Callback(hObject, eventdata, handles)

X8 = str2double(get(hObject,'String'));

if isnan(X8)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if (X8 < 50 || X8 > 1000);
    set(hObject,'String','')
    errordlg('Must be at between 50 and 1000 m/s','Error');
end

function X8_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X8_ButtonDownFcn(hObject, eventdata, handles)
set(handles.X8, 'String', '');

% --- X9 Functions ---
function X9_Callback(hObject, eventdata, handles)

X9 = str2double(get(hObject,'String'));

if isnan(X9)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if (X9 < 0 || X9 > 100);
    set(hObject,'String','')
    errordlg('Must be at between 0 and 100 %','Error');
end

function X9_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X9_ButtonDownFcn(hObject, eventdata, handles)
set(handles.X9, 'String', '');

% --- X10 Functions ---
function X10_Callback(hObject, eventdata, handles)

X10 = str2double(get(hObject,'String'));

if isnan(X10)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if (X10 < 1 || X10 > 60);
    set(hObject,'String','')
    errordlg('Must be at between 1 and 60 min','Error');
end

function X10_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X10_ButtonDownFcn(hObject, eventdata, handles)
```

```
set(handles.X10, 'String', '');

% --- X11 Functions ---
function X11_Callback(hObject, eventdata, handles)

X11 = str2double(get(hObject,'String'));

if isnan(X11)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if (X11 < 0 || X11 > 2000);
    set(hObject,'String','')
    errordlg('Must be at between 0 and 2000 meters','Error');
end

function X11_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X11_ButtonDownFcn(hObject, eventdata, handles)
set(handles.X11, 'String', '');

% --- X12 Functions ---
function X12_Callback(hObject, eventdata, handles)

X12 = str2double(get(hObject,'String'));

if isnan(X12)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if (X12 < 1 || X12 > 14);
    set(hObject,'String','')
    errordlg('Must be at between 1 and 14','Error');
end

function X12_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X12_ButtonDownFcn(hObject, eventdata, handles)
set(handles.X12, 'String', '');

% --- X13 Functions ---
function X13_Callback(hObject, eventdata, handles)

X13 = str2double(get(hObject,'String'));
```

```
if isnan(X13)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if (X13 < 1 || X13 > 60);
    set(hObject,'String','')
    errordlg('Must be at between 1 and 60 min','Error');
end

function X13_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X13_ButtonDownFcn(hObject, eventdata, handles)
set(handles.X13, 'String', '');

% --- X14 Functions ---
function X14_Callback(hObject, eventdata, handles)

X14 = str2double(get(hObject,'String'));

if isnan(X14)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if (X14 < 1 || X14 > 10);
    set(hObject,'String','')
    errordlg('Must be at between 1 and 10','Error');
end

function X14_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X14_ButtonDownFcn(hObject, eventdata, handles)
set(handles.X14, 'String', '');

% GET CONFIDENCE BELIEF FROM USER ON INPUT VARIABLES X1 - X14
% --- Confidence Input, C1, for X1 ---
function C1_Callback(hObject, eventdata, handles)
% Get string from drop down menu box in GUI
C1 = str2double(get(hObject,'String'));

if isnan(C1)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
```

```
if (C1 < 0 || C1 > 100);
    set(hObject,'String','')
    errordlg('Must be at between 0 and 100 %','Error');
end

function C1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function C1_ButtonDownFcn(hObject, eventdata, handles)
set(handles.C1, 'String', '');

% --- Confidence Input, C2, for X2 ---
function C2_Callback(hObject, eventdata, handles)

C2 = str2double(get(hObject,'String'));

if isnan(C2)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if C2 < 0 || C2 > 100;
    set(hObject,'String','')
    errordlg('Must be at between 0 and 100 %','Error');
end

function C2_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function C2_ButtonDownFcn(hObject, eventdata, handles)
set(handles.C2, 'String', '');

% --- Confidence Input, C3, for X3 ---
function C3_Callback(hObject, eventdata, handles)

C3 = str2double(get(hObject,'String'));

if isnan(C3)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if C3 < 0 || C3 > 100;
    set(hObject,'String','')
    errordlg('Must be at between 0 and 100 %','Error');
end

function C3_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function C3_ButtonDownFcn(hObject, eventdata, handles)
set(handles.C3, 'String', '');

% --- Confidence Input, C4, for X4 ---
function C4_Callback(hObject, eventdata, handles)

C4 = str2double(get(hObject,'String'));

if isnan(C4)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if C4 < 0 || C4 > 100;
    set(hObject,'String','')
    errordlg('Must be at between 0 and 100 %','Error');
end

function C4_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function C4_ButtonDownFcn(hObject, eventdata, handles)
set(handles.C4, 'String', '');

% --- Confidence Input, C5, for X5 ---
function C5_Callback(hObject, eventdata, handles)

C5 = str2double(get(hObject,'String'));

if isnan(C5)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if C5 < 0 || C5 > 100;
    set(hObject,'String','')
    errordlg('Must be at between 0 and 100 %','Error');
end

function C5_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function C5_ButtonDownFcn(hObject, eventdata, handles)
set(handles.C5, 'String', '');

% --- Confidence Input, C6, for X6 ---
function C6_Callback(hObject, eventdata, handles)

C6 = str2double(get(hObject,'String'));

if isnan(C6)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if C6 < 0 || C6 > 100;
    set(hObject,'String','')
    errordlg('Must be at between 0 and 100 %','Error');
end

function C6_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function C6_ButtonDownFcn(hObject, eventdata, handles)
set(handles.C6, 'String', '');

% --- Confidence Input, C7, for X7 ---
function C7_Callback(hObject, eventdata, handles)

C7 = str2double(get(hObject,'String'));

if isnan(C7)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if C7 < 0 || C7 > 100;
    set(hObject,'String','')
    errordlg('Must be at between 0 and 100 %','Error');
end

function C7_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function C7_ButtonDownFcn(hObject, eventdata, handles)
set(handles.C7, 'String', '');

% --- Confidence Input, C8, for X8 ---
function C8_Callback(hObject, eventdata, handles)
```

```
C8 = str2double(get(hObject,'String'));

if isnan(C8)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if C8 < 0 || C8 > 100;
    set(hObject,'String','')
    errordlg('Must be at between 0 and 100 %','Error');
end

function C8_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function C8_ButtonDownFcn(hObject, eventdata, handles)
set(handles.C8, 'String', '');

% --- Confidence Input, C9, for X9 ---
function C9_Callback(hObject, eventdata, handles)

C9 = str2double(get(hObject,'String'));

if isnan(C9)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if C9 < 0 || C9 > 100;
    set(hObject,'String','')
    errordlg('Must be at between 0 and 100 %','Error');
end

function C9_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function C9_ButtonDownFcn(hObject, eventdata, handles)
set(handles.C9, 'String', '');

% --- Confidence Input, C10, for X10 ---
function C10_Callback(hObject, eventdata, handles)

C10 = str2double(get(hObject,'String'));

if isnan(C10)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
```

```
if C10 < 0 || C10 > 100;
    set(hObject,'String','')
    errordlg('Must be at between 0 and 100 %','Error');
end

function C10_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function C10_ButtonDownFcn(hObject, eventdata, handles)
set(handles.C10, 'String', '');

% --- Confidence Input, C11, for X11 ---
function C11_Callback(hObject, eventdata, handles)

C11 = str2double(get(hObject,'String'));

if isnan(C11)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if C11 < 0 || C11 > 100;
    set(hObject,'String','')
    errordlg('Must be at between 0 and 100 %','Error');
end

function C11_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function C11_ButtonDownFcn(hObject, eventdata, handles)
set(handles.C11, 'String', '');

% --- Confidence Input, C12, for X12 ---
function C12_Callback(hObject, eventdata, handles)

C12 = str2double(get(hObject,'String'));

if isnan(C12)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if C12 < 0 || C12 > 100;
    set(hObject,'String','')
    errordlg('Must be at between 0 and 100 %','Error');
end

function C12_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function C12_ButtonDownFcn(hObject, eventdata, handles)
set(handles.C12, 'String', '');

% --- Confidence Input, C13, for X13 ---
function C13_Callback(hObject, eventdata, handles)

C13 = str2double(get(hObject,'String'));

if isnan(C13)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if C13 < 0 || C13 > 100;
    set(hObject,'String','')
    errordlg('Must be at between 0 and 100 %','Error');
end

function C13_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function C13_ButtonDownFcn(hObject, eventdata, handles)
set(handles.C13, 'String', '');

% --- Confidence Input, C14, for X14 ---
function C14_Callback(hObject, eventdata, handles)

C14 = str2double(get(hObject,'String'));

if isnan(C14)
    set(hObject,'String','')
    errordlg('Input must be a number','Error');
end
if C14 < 0 || C14 > 100;
    set(hObject,'String','')
    errordlg('Must be at between 0 and 100 %','Error');
end

function C14_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function C14_ButtonDownFcn(hObject, eventdata, handles)
set(handles.C14, 'String', '');

% --- k Functions ---
function K_Callback(hObject, eventdata, handles)
K = str2double(get(hObject,'String'));

function K_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function K_ButtonDownFcn(hObject, eventdata, handles)
set(handles.K, 'String', '');

% --- Executes on button press in User_Data.
function [dstructure] = User_Data_Callback(hObject, eventdata, handles)
% hObject    handle to Load_Scenario (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
data = get(handles.atlas, 'userdata');

dstructure.X1 = str2double(get(handles.X1,'String'));
dstructure.X2 = str2double(get(handles.X2,'String'));
dstructure.X3 = str2double(get(handles.X3,'String'));
dstructure.X4 = str2double(get(handles.X4,'String'));
dstructure.X5 = str2double(get(handles.X5,'String'));
dstructure.X6 = str2double(get(handles.X6,'String'));
dstructure.X7 = str2double(get(handles.X7,'String'));
dstructure.X8 = str2double(get(handles.X8,'String'));
dstructure.X9 = str2double(get(handles.X9,'String'));
dstructure.X10 = str2double(get(handles.X10,'String'));
dstructure.X11 = str2double(get(handles.X11,'String'));
dstructure.X12 = str2double(get(handles.X12,'String'));
dstructure.X13 = str2double(get(handles.X13,'String'));
dstructure.X14 = str2double(get(handles.X14,'String'));
dstructure.C1 = str2double(get(handles.C1,'String'));
dstructure.C2 = str2double(get(handles.C2,'String'));
dstructure.C3= str2double(get(handles.C3,'String'));
dstructure.C4 = str2double(get(handles.C4,'String'));
dstructure.C5 = str2double(get(handles.C5,'String'));
dstructure.C6 = str2double(get(handles.C6,'String'));
dstructure.C7 = str2double(get(handles.C7,'String'));
dstructure.C8 = str2double(get(handles.C8,'String'));
dstructure.C9 = str2double(get(handles.C9,'String'));
dstructure.C10 = str2double(get(handles.C10,'String'));
dstructure.C11 = str2double(get(handles.C11,'String'));
dstructure.C12 = str2double(get(handles.C12,'String'));
dstructure.C13 = str2double(get(handles.C13,'String'));
dstructure.C14 = str2double(get(handles.C14,'String'));
dstructure.K = str2double(get(handles.K,'String'));

data.k = dstructure.K;
```

```
data.dstructure = dstructure;
set(handles.atlas, 'userdata', data);


% --- Executes on button press in Find_Scenario.
function Find_Scenario_Callback(hObject, eventdata, handles)
get (handles.atlas, 'userdata');
[handles] = guiengine(handles,'user_input');
[handles] = guiengine(handles,'load_cases');


% --- Executes on button press in Load_Scenario.
function Load_Scenario_Callback(hObject, eventdata, handles)
get(handles.atlas, 'userdata');
[handles] = guiengine(handles,'load_cases');


% --- Executes on selection change in scenario_listbox.
function scenario_listbox_Callback(hObject, eventdata, handles)
[handles] = guiengine(handles,'user_select_option');


% --- Executes during object creation, after setting all properties.
function scenario_listbox_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --------------------------------------------------------------------
function mnuFile_Callback(hObject, eventdata, handles)
% --------------------------------------------------------------------
function mnuNew_Querry_Callback(hObject, eventdata, handles)
delete(gcf);
DST_ATLAS;
% --------------------------------------------------------------------
function mnuLoad_Database_Callback(hObject, eventdata, handles)
[handles] = guiengine(handles,'DB');

% --- Executes on button press in Load_Data_Base.
function Load_Data_Base_Callback(hObject, eventdata, handles)
[handles] = guiengine(handles,'DB');

function PredKill_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function PredKill_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function PredHelo_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function PredHelo_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in EnemyKilled.
function EnemyKilled_Callback(hObject, eventdata, handles)
[handles] = guiengine(handles,'mean');


% --- Executes on button press in Helo_Flights.
function Helo_Flights_Callback(hObject, eventdata, handles)
[handles] = guiengine(handles,'num-flight');


% --- Executes on button press in DistanceMetric.
function DistanceMetric_Callback(hObject, eventdata, handles)
[handles] = guiengine(handles,'dist-metric');



% --- Executes when selected object is changed in RunSelection.
function RunSelection_SelectionChangeFcn(hObject, eventdata, handles)

% --------------------------------------------------------------------
function mnuExit_Callback(hObject, eventdata, handles)

selection = questdlg('Do you want to close the GUI?',...
                     'Close Request Function',...
                     'Yes','No','Yes');
switch selection,
   case 'Yes',
    delete(gcf)
   case 'No'
     return
end

% --- Executes on button press in Run1.
function Run1_Callback(hObject, eventdata, handles)
[handles] = guiengine(handles,'SetRun1');


% --- Executes on button press in Run2.
function Run2_Callback(hObject, eventdata, handles)
[handles] = guiengine(handles,'SetRun2');


% --- Executes on button press in Run3.
function Run3_Callback(hObject, eventdata, handles)
[handles] = guiengine(handles,'SetRun3');
```

```
% --- Executes on button press in Run4.
function Run4_Callback(hObject, eventdata, handles)
[handles] = guiengine(handles,'SetRun4');


% --- Executes on button press in Run5.
function Run5_Callback(hObject, eventdata, handles)
[handles] = guiengine(handles,'SetRun5');


% --- Executes on button press in AllSqds.
function AllSqds_Callback(hObject, eventdata, handles)
[handles] = guiengine(handles,'SquadAll');


% --- Executes on button press in Sqd1.
function Sqd1_Callback(hObject, eventdata, handles)
 [handles] = guiengine(handles,'Squad1');

% --- Executes on button press in Sqd2.
function Sqd2_Callback(hObject, eventdata, handles)
[handles] = guiengine(handles,'Squad2');

% --- Executes on button press in Sqd3.
function Sqd3_Callback(hObject, eventdata, handles)
[handles] = guiengine(handles,'Squad3');



function X1Var_Callback(hObject, eventdata, handles)
function X1Var_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X2Var_Callback(hObject, eventdata, handles)
function X2Var_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X3Var_Callback(hObject, eventdata, handles)
function X3Var_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X4Var_Callback(hObject, eventdata, handles)
function X4Var_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X5Var_Callback(hObject, eventdata, handles)
function X5Var_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X6Var_Callback(hObject, eventdata, handles)
function X6Var_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X7Var_Callback(hObject, eventdata, handles)
function X7Var_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X8Var_Callback(hObject, eventdata, handles)
function X8Var_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X9Var_Callback(hObject, eventdata, handles)
function X9Var_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X10Var_Callback(hObject, eventdata, handles)
function X10Var_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X11Var_Callback(hObject, eventdata, handles)
function X11Var_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function X12Var_Callback(hObject, eventdata, handles)
function X12Var_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X13Var_Callback(hObject, eventdata, handles)
function X13Var_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function X14Var_Callback(hObject, eventdata, handles)
function X14Var_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

C 3. Function Codes Executed by the GUI (guiengine.m)

As was alluded to in the previous section this code is the computing focal point behind the ATLAS Tool. This code reads in the information entered by the tool user and accesses the knowledge base to calculate the distance metrics and other plots that are a visual aid to the user. This code also incorporates the ANN that was created by JMP from the input-output data from the simulation runs.

```
function [handles] = guiengine(handles,action)
% This is my user defined function where I control my GUI from.

% Here I want to store all userdata that the GUI needs.
data = get(handles.atlas,'userdata');
%
if (isfield(data, 'k') == 1)
      k = data.k;
end;

% All these actions are user defined. String names are arbitrary.
switch action

    case 'initialize'
        axes(handles.Icon);
        picture = imread('Pics\ATLAS2.jpg');
        image(picture);
```

```
        axis off;

        axes(handles.Graph);
        picture1 = imread('Pics\Graphs.jpg');
        image(picture1);
        axis off;

        axes(handles.Resupply);
        picture2 = imread('Pics\Reorder_Point.jpg');
        image(picture2);
        axis off;

        % Initialize to have no run and no squads selected.
        data.run = -1;
        data.squad = -1;
%         data.CaseNum = 0;

    case 'SquadAll'
        data.squad = 0;
    case 'Squad1'
        data.squad = 1;
    case 'Squad2'
        data.squad = 2;
    case 'Squad3'
        data.squad = 3;
    case 'SetRun1'
        data.run = 1;
    case 'SetRun2'
        data.run = 2;
    case 'SetRun3'
        data.run = 3;
    case 'SetRun4'
        data.run = 4;
    case 'SetRun5'
        data.run = 5;

    case 'DB'
        data.DB = xlsread('Scenario Data.xls',-1);

    case 'user_input'
        data.Request_from_user = Find_Scenarios(data, handles, data.DB,
data.dstructure);

    case 'load_cases'
        data.Load_Cases = LoadCases(data, handles);


    case 'user_select_option'
        option_selected = get(handles.scenario_listbox, 'Value');
        CaseList = str2num(get(handles.scenario_listbox, 'String'));
        try %#ok<TRYNC>
            data.CaseNum = CaseList(option_selected);
        end;

    case 'mean'
```

```
        warning off %#ok<WNOFF>

        axes(handles.Graph);
        results_in = xlsread('User Output.xls');
        ListStr = results_in(1:k,1);
        x_axis = ListStr;
        y_axis = results_in(1:k,2);
        error = results_in(1:k,3);
        errorbar(x_axis,y_axis,error,'sr',...
            'LineWidth',0,...
            'MarkerEdgeColor','k',...
            'MarkerFaceColor','r',...
            'MarkerSize',8)
%         axis([0 1010 0 1.1])
        axis tight
        title 'Percent Enemy Killed by Case Number'
        xlabel('Case Number');
        ylabel('Percent Enenmy Killed');
        grid on

        warning on %#ok<WNON>

    case 'num-flight'
        axes(handles.Graph);
        results_in = xlsread('User Output.xls');
        ListStr = results_in(1:k,1);
        for i=1:k
            for z=1:5
                f1='C:\Documents and Settings\gtg730n\My
Documents\MATLAB\Helo Flights in Days - 3 sqds\Helo_-0-';
%                 f1='G:\MATLAB\Helo Flights in Days - 3 sqds\Helo_-0-
';
                f2=ListStr(i,1);
                f2_=num2str(f2);
                f3='_pos_';
                f4=z;
                f4_=num2str(f4);
                f5='.csv';
                file = strcat(f1,f2_,f3,f4_,f5);

RR=textread(file,'','delimiter',',','commentstyle','shell');

                % Squad 1
                S1=sum(RR(:,2)==1);
                A(z,1)=S1; %#ok<AGROW>

                % Squad 2
                S2=sum(RR(:,2)==2);
                B(z,1)=S2; %#ok<AGROW>

                % Squad 3
                S3=sum(RR(:,2)==3);
                C(z,1)=S3; %#ok<AGROW>
            end
```

```
            s1_flt = round((sum(A)/5) + .49);
            s2_flt = round((sum(B)/5) + .49);
            s3_flt = round((sum(C)/5) + .49);
            Y(i,:) = [s1_flt s2_flt s3_flt]; %#ok<AGROW>
        end
        Helo = Y(1:k,1:3);

        bar(Helo,'stack');
        axis tight;
        title 'Resupply Flights by Squad';
        xlabel('Case Number');
        ylabel({'Amount of Resupply Helos Per Squad'});
        legend('Sqd 1', 'Sqd 2', 'Sqd 3','Location','Best');
        grid on

    case 'dist-metric'
        axes(handles.Graph);
        results_in = xlsread('User Output.xls');
        ListStr = results_in(1:k,1);

        x_axis = ListStr;
        y1_axis = results_in(1:k,5);
        y2_axis = results_in(1:k,7);

        ed = [x_axis y1_axis];
        ED = sortrows(ed,1);
        ED1= sortrows(ed,2);

        oec = [x_axis y2_axis];
        OEC = sortrows(oec,1);
        OEC1 = sortrows(oec,2);

        % Creates one plot with two y axis
        [XA,Y1,Y2] = plotyy(ED(:,1),ED(:,2),OEC(:,1),OEC(:,2));
        set(Y1,'LineStyle',':',...
            'Marker','s',...
            'LineWidth',2,...
            'MarkerEdgeColor','k',...
            'MarkerFaceColor','r',...
            'MarkerSize',8)
        set(Y2,'LineStyle',':',...
            'Marker','d',...
            'LineWidth',2,...
            'MarkerEdgeColor','k',...
            'MarkerFaceColor','g',...
            'MarkerSize',8);
        set(get(XA(1),'Ylabel'),'String','Euclidean Distance')
        set(get(XA(2),'Ylabel'),'String','OEC with Weights')
        if ED1(k,2) > OEC1(k,2)
            ymax=ED1(k,2) + 25; %#ok<NASGU>
        else
            ymax=OEC1(k,2) + 25; %#ok<NASGU>
        end
        if ED1 (1,2) < OEC1(1,2)
            ymin=ED1(1,2) - 25; %#ok<NASGU>
        else
```

```
            ymin=OEC1(1,2) - 25 ; %#ok<NASGU>
        end
%        axis([0 1000 ymin ymax])
        title 'Distance Metrics from Entered Data'
        xlabel('Case Number');
        set (get(XA(1),'Ylabel'),'String','Euclidean Distance');
        set (get(XA(2),'Ylabel'),'String','OEC with Weighting');
        legend([Y1, Y2],'Euclidean Distance','OEC','Location','Best');
        grid on

%  Creates two plots on the same figure
%         axes(handles.Graph);
%         subplot(2,1,1);
%         plot(ED(:,1),ED(:,2),':sr','LineWidth',0.5,...
%             'MarkerEdgeColor','k',...
%             'MarkerFaceColor','r',...
%             'MarkerSize',10)
%         axis tight;
%         title 'Euclidean Distance from Entered Data'
%         xlabel('Case Number');
%         ylabel({'Euclidean Distance'});
%         grid on;
%
%         subplot(2,1,2);
%         plot(OEC(:,1),OEC(:,2),'-.db','LineWidth',0.5,...
%             'MarkerEdgeColor','k',...
%             'MarkerFaceColor','b',...
%             'MarkerSize',10)
%         axis tight;
%         title 'OEC with User Weights'
%         xlabel('Case Number');
%         ylabel({'OEC Value'});
%         grid on;
%         axis tight;
    case 'PlotResupply'
end

set(handles.atlas,'userdata',data);

%% Additional Function Calls

function Request_from_user = Find_Scenarios(data, handles, DB,
dstructure) %#ok<INUSL>

dim = size(DB,1);

X1 = round(dstructure.X1*2880);
X2 = dstructure.X2*2880;
X3 = dstructure.X3*2;
X4 = dstructure.X4;
X5 = dstructure.X5;
X6 = dstructure.X6*2;
X7 = dstructure.X7*2;
X8 = dstructure.X8;
X9 = dstructure.X9;
X10 = dstructure.X10*2;
```

```
X11 = dstructure.X11/30;
X12 = dstructure.X12;
X13 = dstructure.X13*2;
X14 = dstructure.X14;

c1 = dstructure.C1;
c2 = dstructure.C2;
c3 = dstructure.C3;
c4 = dstructure.C4;
c5 = dstructure.C5;
c6 = dstructure.C6;
c7 = dstructure.C7;
c8 = dstructure.C8;
c9 = dstructure.C9;
c10 = dstructure.C10;
c11 = dstructure.C11;
c12 = dstructure.C12;
c13 = dstructure.C13;
c14 = dstructure.C14;

k = data.k;

% Execution and evaluation of uncertainty from User via Fuzzy Inference
% System (FIS)

% Open FIS for each variable
x1 = readfis('X1');
x2 = readfis('X2');
x3 = readfis('X3');
x4 = readfis('X4');
x5 = readfis('X5');
x6 = readfis('X6');
x7 = readfis('X7');
x8 = readfis('X8');
x9 = readfis('X9');
x10 = readfis('X10');
x11 = readfis('X11');
x12 = readfis('X12');
x13 = readfis('X13');
x14 = readfis('X14');

% Evaluate User entry through FIS
C_(1,1)=evalfis(c1,x1);
C_(1,2)=evalfis(c2,x2);
C_(1,3)=evalfis(c3,x3);
C_(1,4)=evalfis(c4,x4);
C_(1,5)=evalfis(c5,x5);
C_(1,6)=evalfis(c6,x6);
C_(1,7)=evalfis(c7,x7);
C_(1,8)=evalfis(c8,x8);
C_(1,9)=evalfis(c9,x9);
C_(1,10)=evalfis(c10,x10);
C_(1,11)=evalfis(c11,x11);
C_(1,12)=evalfis(c12,x12);
C_(1,13)=evalfis(c13,x13);
C_(1,14)=evalfis(c14,x14);
```

```
% Calculate Normalized Weights for OEC
C=C_/sum(C_);


% User Input Variables
Q = [X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14];

for i = 1:dim
    % Calculate Euclidean Distance from Case Entered
    enemy(i,1) = sqrt((DB(i,11)-Q(1,11))^2 + (DB(i,12)-Q(1,12))^2 +
(DB(i,13)-Q(1,13))^2 + (DB(i,14)-Q(1,14))^2); %#ok<AGROW>
end

% Calculate Predicted % Enemy Killed and Helicopter Flights Needed to
Conduct the Mission
PK = (10.024002644252 + -0.971958936043124 * 1/(1+ exp(-1*((-
18.7116487108048) + -9.93632898683372 * ((X1 -
7205.65047518479)/4139.23541383032)...
    + -0.111224178706112 * ((X2 - 7302.14677930306)/4107.52040583522) +
0.129210273235375 * ((X3 - 723.970432946146)/414.978989218406)...
    + -0.0871358113104067 * ((X4 - 24.7233368532207/14.484500510097) +
0.222860815617708 * ((X5 - 25.1795142555438)/14.5303314230908)...
    + 0.257795254987707 * ((X6 - 361.198521647307)/207.644905153702) +
-0.276076427821489 * ((X7 - 60.1636747624076)/34.8320893825286)...
    + -0.00419738760839964 * ((X8 - 523.460401267159) /
276.748638566215) + 0.0763190499706875 * ((X9 - 50.104540654699) /
28.755046445147)...
    + -0.0253727280455679 * ((X10 - 60.0179514255544) /
34.548563956929) + -0.0618174920686842 * ((X11 - 33.3167898627244) /
19.3328736408942)...
    + 0.0493732312494361 * ((X12 - 7.24181626187962) /
3.90175653998081) + 0.105456479003555 * ((X13 - 60.7286166842661) /
34.5003179580264)...
    + 0.0115968031037884 * ((X14 - 5.00527983104541) /
2.91002742604659))))) + -0.0156376601574198 * 1/(1+ exp(-1*((-
2.09221815819687)...
    + 4.48025237502459 * ((X1 - 7205.65047518479)/4139.23541383032) +
0.918282532903091 * ((X2 - 7302.14677930306)/4107.52040583522)...
    + -4.19354575138397 * ((X3 - 723.970432946146)/414.978989218406) +
-2.56439387825574 * ((X4 - 24.7233368532207)/14.484500510097)...
    + 1.84441776990581 * ((X5 - 25.1795142555438)/14.5303314230908) +
6.35366808197846 * ((X6 - 361.198521647307)/207.644905153702)...
    + 2.5862789729347 * ((X7 - 60.1636747624076)/34.8320893825286) +
12.6868223006802 * ((X8 - 523.460401267159)/276.748638566215)...
    + 1.52106129998419 * ((X9 - 50.104540654699)/28.755046445147) + -
7.98366669203337 * ((X10 - 60.0179514255544)/34.548563956929)...
    + 1.83803947347364 * ((X11 - 33.3167898627244) /19.3328736408942) +
-3.53429526314043 * ((X12 - 7.24181626187962) /3.90175653998081)...
    + 2.43123493814063 * ((X13 - 60.7286166842661) /34.5003179580264) +
-3.86471557336723 * ((X14 - 5.00527983104541) /2.91002742604659))))...
    + -4.27711839397722 * 1/(1+ exp(-1*(11.7668455339837 + -
0.0902967830802342 * ((X1 - 7205.65047518479) /4139.23541383032)...
    + 5.95727196609715 * ((X2 - 7302.14677930306) /4107.52040583522) +
0.10361045517173 * ((X3 - 723.970432946146) /414.978989218406)...
```

```
    + 0.0395481092137692 * ((X4 - 24.7233368532207) /14.484500510097) +
-0.00407363207420756 * ((X5 - 25.1795142555438) / 14.5303314230908)...
    + 0.0340514035582536 * ((X6 - 361.198521647307) / 207.644905153702)
+ -0.00632427841118012 * ((X7 -60.1636747624076) / 34.8320893825286)...
    + -0.444303118494556 * ((X8 -523.460401267159) / 276.748638566215)
+ 0.0757763321583199 * ((X9 -50.104540654699) / 28.755046445147)...
    + 0.0416087612093763 * ((X10 -60.0179514255544) / 34.548563956929)
+ -0.0841150837874823 * ((X11 - 33.3167898627244) /
19.3328736408942)...
    + 0.0534961829625253 * ((X12 - 7.24181626187962) /
3.90175653998081) + -0.0762039730505856 * ((X13 - 60.7286166842661) /
34.5003179580264)...
    + 0.0162429789025519 *((X14 - 5.00527983104541) /
2.91002742604659)))) + -6.10669851425778 * 1/(1+ exp(-
1*(5.64078979538522...
    + 0.00359131215482465 * ((X1 - 7205.65047518479) /4139.23541383032)
+ -0.065892707676202 * ((X2 - 7302.14677930306) /4107.52040583522)...
    + -0.334877223152337 * ((X3 - 723.970432946146) /414.978989218406)
+ -0.0510557864906132 * ((X4 - 24.7233368532207) / 14.484500510097)...
    + 0.0555239335756663 * ((X5 - 25.1795142555438) / 14.5303314230908)
+ -0.102011513453141 * ((X6 - 361.198521647307) / 207.644905153702)...
    + 0.197290497101268 * ((X7 - 60.1636747624076) / 34.8320893825286)
+ 0.305992639026365 * ((X8 - 523.460401267159) / 276.748638566215)...
    + -0.0416136823008085 * ((X9 - 50.104540654699) / 28.755046445147)
+ -0.0756398283433795 * ((X10 - 60.0179514255544) / 34.548563956929)...
    + 0.156427875701199 * ((X11 - 33.3167898627244) / 19.3328736408942)
+ 1.79518397171281 * ((X12 - 7.24181626187962) / 3.90175653998081)...
    + 0.0404379799100313 * ((X13 - 60.7286166842661) /
34.5003179580264) + -0.358975139184959 * ((X14 -5.00527983104541) /
2.91002742604659))))...
    + 6.63921048304502 * 1/(1+ exp(-1*(1.26168159961907 +
0.0882654281012707 * ((X1 - 7205.65047518479) /4139.23541383032)...
    + -0.022581882089312 * ((X2 - 7302.14677930306) /4107.52040583522)
+ 0.105179866630451 * ((X3 - 723.970432946146) /414.978989218406)...
    + 0.024148075937347 * ((X4 - 24.7233368532207) /14.484500510097) +
-0.0734284755932574 * ((X5 - 25.1795142555438) /14.5303314230908)...
    + 0.087505611408095 * ((X6 - 361.198521647307) /207.644905153702) +
0.0741988341005277 * ((X7 - 60.1636747624076) /34.8320893825286)...
    + 3.52257777062392 * ((X8 - 523.460401267159) /276.748638566215) +
0.0711396627565711 * ((X9 - 50.104540654699) /28.755046445147)...
    + -0.0972060435654332 * ((X10 - 60.0179514255544) /
34.548563956929) + 0.102672348119397 * ((X11 - 33.3167898627244) /
19.3328736408942)...
    + 0.130771671358677 * ((X12 - 7.24181626187962) / 3.90175653998081)
+ -0.139681704261056 * ((X13 - 60.7286166842661) / 34.5003179580264)...
    + 0.0236549647484934 * ((X14 -5.00527983104541) /
2.91002742604659)))) + -0.585510697852289 * 1/(1+ exp(-1*((-
4.68258036209264)...
    + -2.5239424712346 * ((X1 - 7205.65047518479) /4139.23541383032) +
0.150087475733876 * ((X2 - 7302.14677930306) /4107.52040583522)...
    + -2.53284801576807 * ((X3 - 723.970432946146) /414.978989218406) +
-0.290447735621779 * ((X4 - 24.7233368532207) /14.484500510097)...
    + 0.493779296946835 * ((X5 - 25.1795142555438) /14.5303314230908) +
-1.01491283957493 * ((X6 - 361.198521647307) /207.644905153702)...
    + -0.536309696410602 * ((X7 - 60.1636747624076) /34.8320893825286)
+ 0.17424731206661 * ((X8 - 523.460401267159) /276.748638566215)...
```

```
    + 0.38249914401023 * ((X9 - 50.104540654699) /28.755046445147) +
0.354045052417262 * ((X10 - 60.0179514255544) /34.548563956929)...
    + -0.129600056245359 * ((X11 - 33.3167898627244) /19.3328736408942)
+ -0.43828275493555 * ((X12 - 7.24181626187962) /3.90175653998081)...
    + 0.274352641123028 * ((X13 - 60.7286166842661) /34.5003179580264)
+ 0.0919416155651336 * ((X14 - 5.00527983104541) /
2.91002742604659))))...
    + 4.84431550905819 * 1/(1+ exp(-1*((-0.879269090341992) + -
0.0470496718465418 * ((X1 - 7205.65047518479) / 4139.23541383032)...
    + 0.0603038182066098 * ((X2 - 7302.14677930306) / 4107.52040583522)
+ -0.440762913462218 * ((X3 - 723.970432946146) / 414.978989218406)...
    + -0.0544658750683794 * ((X4 -24.7233368532207) / 14.484500510097)
+ 0.0920820188155522 * ((X5 -25.1795142555438) / 14.5303314230908)...
    + -0.12040930678622 * ((X6 -361.198521647307) / 207.644905153702) +
-0.123354377326387 * ((X7 -60.1636747624076) / 34.8320893825286)...
    + -3.4772154442584 * ((X8 -523.460401267159) / 276.748638566215) +
0.00511933235499323 * ((X9 - 50.104540654699) / 28.755046445147)...
    + 0.137340277281053 * ((X10 - 60.0179514255544) / 34.548563956929)
+ -0.122806512356034 * ((X11 - 33.3167898627244) / 19.3328736408942)...
    + -0.169852657438889 * ((X12 - 7.24181626187962) /
3.90175653998081) + 0.206906060517658 * ((X13 - 60.7286166842661) /
34.5003179580264)...
    + -0.0357812831437477 * ((X14 - 5.00527983104541) /
2.91002742604659)))) + -2.89544533608445 * 1/(1+ exp(-
1*(1.43294199872541...
    + -0.477597116318321 * ((X1 - 7205.65047518479) /4139.23541383032)
+ 0.344447910975749 * ((X2 - 7302.14677930306) /4107.52040583522)...
    + 0.0545818513590741 * ((X3 - 723.970432946146) /414.978989218406)
+ 0.338426109475185 * ((X4 - 24.7233368532207) /14.484500510097)...
    + 0.150812012246556 * ((X5 - 25.1795142555438) /14.5303314230908) +
-0.10831214937393 * ((X6 - 361.198521647307) /207.644905153702)...
    + 0.0259190175958278 * ((X7 - 60.1636747624076) /34.8320893825286)
+ 0.68889030090604 * ((X8 - 523.460401267159) /276.748638566215)...
    + -0.0208810140346043 * ((X9 - 50.104540654699) /28.755046445147) +
0.0686555637911968 * ((X10 - 60.0179514255544) /34.548563956929)...
    + -0.0457715789922175 * ((X11 - 33.3167898627244) /
19.3328736408942) + 0.47484073949593 * ((X12 - 7.24181626187962) /
3.90175653998081)...
    + 0.0720773663391528 * ((X13 - 60.7286166842661) /
34.5003179580264) + 1.56622228493644 * ((X14 - 5.00527983104541) /
2.91002742604659))))...
    + -2.12751010742092 * 1/(1+ exp(-1*(1.81299776510915 +
0.0441795647587806 * ((X1 - 7205.65047518479) /4139.23541383032)...
    + -0.0970696777858851 * ((X2 - 7302.14677930306) /
4107.52040583522) + -0.262794093733253 * ((X3 - 723.970432946146) /
414.978989218406)...
    + 0.0439872444522603 * ((X4 - 24.7233368532207) / 14.484500510097)
+ -0.0305438092430749 * ((X5 - 25.1795142555438) / 14.5303314230908)...
    + 0.122161927022276 * ((X6 - 361.198521647307) / 207.644905153702)
+ 0.0419181798417392 * ((X7 - 60.1636747624076) / 34.8320893825286)...
    + 3.12156227204938 * ((X8 - 523.460401267159) / 276.748638566215) +
0.217616319837778 * ((X9 - 50.104540654699) / 28.755046445147)...
    + -0.106845887243196 * ((X10 - 60.0179514255544) / 34.548563956929)
+ 0.0826485752021084 * ((X11 - 33.3167898627244) / 19.3328736408942)...
    + 0.131647917501276 * ((X12 - 7.24181626187962) / 3.90175653998081)
+ -0.0629011742662475 * ((X13 -60.7286166842661) / 34.5003179580264)...
```

```
    + 0.00947213222242092 * ((X14 - 5.00527983104541) /
2.91002742604659)))) + -2.42036633869115 * 1/(1+ exp(-
1*(5.43810756923561...
    + 0.528927389974741 * ((X1 - 7205.65047518479) /4139.23541383032) +
-0.032938425458737 * ((X2 - 7302.14677930306) /4107.52040583522)...
    + 2.70696901108989 * ((X3 - 723.970432946146) /414.978989218406) +
-0.0579523964920225 * ((X4 - 24.7233368532207) / 14.484500510097)...
    + -0.31132332316452 * ((X5 - 25.1795142555438) / 14.5303314230908)
+ 0.272189622464364 * ((X6 - 361.198521647307) / 207.644905153702)...
    + -0.053767720840101 * ((X7 - 60.1636747624076) / 34.8320893825286)
+ 0.590455161096687 * ((X8 - 523.460401267159) / 276.748638566215)...
    + -0.311063738810189 * ((X9 - 50.104540654699) / 28.755046445147) +
-0.023866771543419 * ((X10 - 60.0179514255544) / 34.548563956929)...
    + -0.086144513351825 * ((X11 - 33.3167898627244) /
19.3328736408942) + 0.239387255449479 * ((X12 - 7.24181626187962) /
3.90175653998081)...
    + -0.0143407048413802 * ((X13 -60.7286166842661) /
34.5003179580264) + -0.0738218394270471 * ((X14 - 5.00527983104541) /
2.91002742604659)))))...
    * 27.3336495273715 + 32.4649947201689;

set(handles.PredKill, 'String', PK)


PH = (23.6141005087954 + 13.8275871669317 * 1/(1+ exp(-1*((-
18.7116487108048) + -9.93632898683372 * ((X1 -
7205.65047518479)/4139.23541383032)...
    + -0.111224178706112 * ((X2 - 7302.14677930306)/4107.52040583522) +
0.129210273235375 * ((X3 - 723.970432946146)/414.978989218406)...
    + -0.0871358113104067 * ((X4 - 24.7233368532207/14.484500510097) +
0.222860815617708 * ((X5 - 25.1795142555438)/14.5303314230908)...
    + 0.257795254987707 * ((X6 - 361.198521647307)/207.644905153702) +
-0.276076427821489 * ((X7 - 60.1636747624076)/34.8320893825286)...
    + -0.00419738760839964 * ((X8 - 523.460401267159) /
276.748638566215) + 0.0763190499706875 * ((X9 - 50.104540654699) /
28.755046445147)...
    + -0.0253727280455679 * ((X10 - 60.0179514255544) /
34.548563956929) + -0.0618174920686842 * ((X11 - 33.3167898627244) /
19.3328736408942)...
    + 0.0493732312494361 * ((X12 - 7.24181626187962) /
3.90175653998081) + 0.105456479003555 * ((X13 - 60.7286166842661) /
34.5003179580264)...
    + 0.0115968031037884 * ((X14 - 5.00527983104541) /
2.91002742604659))))) + -0.339890141871457 * 1/(1+ exp(-1*((-
2.09221815819687)...
    + 4.48025237502459 * ((X1 - 7205.65047518479)/4139.23541383032) +
0.918282532903091 * ((X2 - 7302.14677930306)/4107.52040583522)...
    + -4.19354575138397 * ((X3 - 723.970432946146)/414.978989218406) +
-2.56439387825574 * ((X4 - 24.7233368532207)/14.484500510097)...
    + 1.84441776990581 * ((X5 - 25.1795142555438)/14.5303314230908) +
6.35366808197846 * ((X6 - 361.198521647307)/207.644905153702)...
    + 2.5862789729347 * ((X7 - 60.1636747624076)/34.8320893825286) +
12.6868223006802 * ((X8 - 523.460401267159)/276.748638566215)...
    + 1.52106129998419 * ((X9 - 50.104540654699)/28.755046445147) + -
7.98366669203337 * ((X10 - 60.0179514255544)/34.548563956929)...
```

```
    + 1.83803947347364 * ((X11 - 33.3167898627244) /19.3328736408942) +
-3.53429526314043 * ((X12 - 7.24181626187962) /3.90175653998081)...
    + 2.43123493814063 * ((X13 - 60.7286166842661) /34.5003179580264) +
-3.86471557336723 * ((X14 - 5.00527983104541) /2.91002742604659))))...
    + -15.2128423597793 * 1/(1+ exp(-1*(11.7668455339837 + -
0.0902967830802342 * ((X1 - 7205.65047518479) /4139.23541383032)...
    + 5.95727196609715 * ((X2 - 7302.14677930306) /4107.52040583522) +
0.10361045517173 * ((X3 - 723.970432946146) /414.978989218406)...
    + 0.0395481092137692 * ((X4 - 24.7233368532207) /14.484500510097) +
-0.00407363207420756 * ((X5 - 25.1795142555438) / 14.5303314230908)...
    + 0.0340514035582536 * ((X6 - 361.198521647307) / 207.644905153702)
+ -0.00632427841118012 * ((X7 -60.1636747624076) / 34.8320893825286)...
    + -0.444303118494556 * ((X8 -523.460401267159) / 276.748638566215)
+ 0.0757763321583199 * ((X9 -50.104540654699) / 28.755046445147)...
    + 0.0416087612093763 * ((X10 -60.0179514255544) / 34.548563956929)
+ -0.0841150837874823 * ((X11 - 33.3167898627244) /
19.3328736408942)...
    + 0.0534961829625253 * ((X12 - 7.24181626187962) /
3.90175653998081) + -0.0762039730505856 * ((X13 - 60.7286166842661) /
34.5003179580264)...
    + 0.0162429789025519 *((X14 - 5.00527983104541) /
2.91002742604659))))+ -0.353174933924453 * 1/(1+ exp(-
1*(5.64078979538522...
    + 0.00359131215482465 * ((X1 - 7205.65047518479) /4139.23541383032)
+ -0.065892707676202 * ((X2 - 7302.14677930306) /4107.52040583522)...
    + -0.334877223152337 * ((X3 - 723.970432946146) /414.978989218406)
+ -0.0510557864906132 * ((X4 - 24.7233368532207) / 14.484500510097)...
    + 0.0555239335756663 * ((X5 - 25.1795142555438) / 14.5303314230908)
+ -0.102011513453141 * ((X6 - 361.198521647307) / 207.644905153702)...
    + 0.197290497101268 * ((X7 - 60.1636747624076) / 34.8320893825286)
+ 0.305992639026365 * ((X8 - 523.460401267159) / 276.748638566215)...
    + -0.0416136823008085 * ((X9 - 50.104540654699) / 28.755046445147)
+ -0.0756398283433795 * ((X10 - 60.0179514255544) / 34.548563956929)...
    + 0.156427875701199 * ((X11 - 33.3167898627244) / 19.3328736408942)
+ 1.79518397171281 * ((X12 - 7.24181626187962) / 3.90175653998081)...
    + 0.0404379799100313 * ((X13 - 60.7286166842661) /
34.5003179580264) + -0.358975139184959 * ((X14 -5.00527983104541) /
2.91002742604659))))...
    + -17.0460845945044 * 1/(1+ exp(-1*(1.26168159961907 +
0.0882654281012707 * ((X1 - 7205.65047518479) /4139.23541383032)...
    + -0.022581882089312 * ((X2 - 7302.14677930306) /4107.52040583522)
+ 0.105179866630451 * ((X3 - 723.970432946146) /414.978989218406)...
    + 0.024148075937347 * ((X4 - 24.7233368532207) /14.484500510097) +
-0.0734284755932574 * ((X5 - 25.1795142555438) /14.5303314230908)...
    + 0.087505611408095 * ((X6 - 361.198521647307) /207.644905153702) +
0.0741988341005277 * ((X7 - 60.1636747624076) /34.8320893825286)...
    + 3.52257777062392 * ((X8 - 523.460401267159) /276.748638566215) +
0.0711396627565711 * ((X9 - 50.104540654699) /28.755046445147)...
    + -0.0972060435654332 * ((X10 - 60.0179514255544) /
34.548563956929) + 0.102672348119397 * ((X11 - 33.3167898627244) /
19.3328736408942)...
    + 0.130771671358677 * ((X12 - 7.24181626187962) / 3.90175653998081)
+ -0.139681704261056 * ((X13 - 60.7286166842661) / 34.5003179580264)...
    + 0.0236549647484934 * ((X14 -5.00527983104541) /
2.91002742604659))))+ -0.552047701867046 * 1/(1+ exp(-1*((-
4.68258036209264)...
```

```
    + -2.5239424712346 * ((X1 - 7205.65047518479) /4139.23541383032) +
0.150087475733876 * ((X2 - 7302.14677930306) /4107.52040583522)...
    + -2.53284801576807 * ((X3 - 723.970432946146) /414.978989218406) +
-0.290447735621779 * ((X4 - 24.7233368532207) /14.484500510097)...
    + 0.493779296946835 * ((X5 - 25.1795142555438) /14.5303314230908) +
-1.01491283957493 * ((X6 - 361.198521647307) /207.644905153702)...
    + -0.536309696410602 * ((X7 - 60.1636747624076) /34.8320893825286)
+ 0.17424731206661 * ((X8 - 523.460401267159) /276.748638566215)...
    + 0.38249914401023 * ((X9 - 50.104540654699) /28.755046445147) +
0.354045052417262 * ((X10 - 60.0179514255544) /34.548563956929)...
    + -0.129600056245359 * ((X11 - 33.3167898627244) /19.3328736408942)
+ -0.43828275493555 * ((X12 - 7.24181626187962) /3.90175653998081)...
    + 0.274352641123028 * ((X13 - 60.7286166842661) /34.5003179580264)
+ 0.0919416155651336 * ((X14 - 5.00527983104541) /
2.91002742604659))))...
    + -6.79711169996982 * 1/(1+ exp(-1*((-0.879269090341992) + -
0.0470496718465418 * ((X1 - 7205.65047518479) / 4139.23541383032)...
    + 0.0603038182066098 * ((X2 - 7302.14677930306) / 4107.52040583522)
+ -0.440762913462218 * ((X3 - 723.970432946146) / 414.978989218406)...
    + -0.0544658750683794 * ((X4 -24.7233368532207) / 14.484500510097)
+ 0.0920820188155522 * ((X5 -25.1795142555438) / 14.5303314230908)...
    + -0.12040930678622 * ((X6 -361.198521647307) / 207.644905153702) +
-0.123354377326387 * ((X7 -60.1636747624076) / 34.8320893825286)...
    + -3.4772154442584 * ((X8 -523.460401267159) / 276.748638566215) +
0.00511933235499323 * ((X9 - 50.104540654699) / 28.755046445147)...
    + 0.137340277281053 * ((X10 - 60.0179514255544) / 34.548563956929)
+ -0.122806512356034 * ((X11 - 33.3167898627244) / 19.3328736408942)...
    + -0.169852657438889 * ((X12 - 7.24181626187962) /
3.90175653998081) + 0.206906060517658 * ((X13 - 60.7286166842661) /
34.5003179580264)...
    + -0.0357812831437477 * ((X14 - 5.00527983104541) /
2.91002742604659))))+ -0.030434410951668 * 1/(1+ exp(-
1*(1.43294199872541...
    + -0.477597116318321 * ((X1 - 7205.65047518479) /4139.23541383032)
+ 0.344447910975749 * ((X2 - 7302.14677930306) /4107.52040583522)...
    + 0.0545818513590741 * ((X3 - 723.970432946146) /414.978989218406)
+ 0.338426109475185 * ((X4 - 24.7233368532207) /14.484500510097)...
    + 0.150812012246556 * ((X5 - 25.1795142555438) /14.5303314230908) +
-0.10831214937393 * ((X6 - 361.198521647307) /207.644905153702)...
    + 0.0259190175958278 * ((X7 - 60.1636747624076) /34.8320893825286)
+ 0.68889030090604 * ((X8 - 523.460401267159) /276.748638566215)...
    + -0.0208810140346043 * ((X9 - 50.104540654699) /28.755046445147) +
0.0686555637911968 * ((X10 - 60.0179514255544) /34.548563956929)...
    + -0.0457715789922175 * ((X11 - 33.3167898627244) /
19.3328736408942) + 0.47484073949593 * ((X12 - 7.24181626187962) /
3.90175653998081)...
    + 0.0720773663391528 * ((X13 - 60.7286166842661) /
34.5003179580264) + 1.56622228493644 * ((X14 - 5.00527983104541) /
2.91002742604659))))...
    + 10.7259322156784 * 1/(1+ exp(-1*(1.81299776510915 +
0.0441795647587806 * ((X1 - 7205.65047518479) /4139.23541383032)...
    + -0.0970696777858851 * ((X2 - 7302.14677930306) /
4107.52040583522) + -0.262794093733253 * ((X3 - 723.970432946146) /
414.978989218406)...
    + 0.0439872444522603 * ((X4 - 24.7233368532207) / 14.484500510097)
+ -0.0305438092430749 * ((X5 - 25.1795142555438) / 14.5303314230908)...
```

```
    + 0.122161927022276 * ((X6 - 361.198521647307) / 207.644905153702)
+ 0.0419181798417392 * ((X7 - 60.1636747624076) / 34.8320893825286)...
    + 3.12156227204938 * ((X8 - 523.460401267159) / 276.748638566215) +
0.217616319837778 * ((X9 - 50.104540654699) / 28.755046445147)...
    + -0.106845887243196 * ((X10 - 60.0179514255544) / 34.548563956929)
+ 0.0826485752021084 * ((X11 - 33.3167898627244) / 19.3328736408942)...
    + 0.131647917501276 * ((X12 - 7.24181626187962) / 3.90175653998081)
+ -0.0629011742662475 * ((X13 -60.7286166842661) / 34.5003179580264)...
    + 0.00947213222242092 * ((X14 - 5.00527983104541) /
2.91002742604659))))+ -2.11357729109717 * 1/(1+ exp(-
1*(5.43810756923561...
    + 0.528927389974741 * ((X1 - 7205.65047518479) /4139.23541383032) +
-0.032938425458737 * ((X2 - 7302.14677930306) /4107.52040583522)...
    + 2.70696901108989 * ((X3 - 723.970432946146) /414.978989218406) +
-0.0579523964920225 * ((X4 - 24.7233368532207) / 14.484500510097)...
    + -0.31132332316452 * ((X5 - 25.1795142555438) / 14.530331423090８)
+ 0.272189622464364 * ((X6 - 361.198521647307) / 207.644905153702)...
    + -0.053767720840101 * ((X7 - 60.1636747624076) / 34.8320893825286)
+ 0.590455161096687 * ((X8 - 523.460401267159) / 276.748638566215)...
    + -0.311063738810189 * ((X9 - 50.104540654699) / 28.755046445147) +
-0.023866771543419 * ((X10 - 60.0179514255544) / 34.548563956929)...
    + -0.086144513351825 * ((X11 - 33.3167898627244) /
19.3328736408942) + 0.239387255449479 * ((X12 - 7.24181626187962) /
3.90175653998081)...
    + -0.0143407048413802 * ((X13 -60.7286166842661) /
34.5003179580264) + -0.0738218394270471 * ((X14 - 5.00527983104541) /
2.91002742604659)))))...
    * 33.6599984047734 + 34.5860612460401;

set(handles.PredHelo, 'String', PH)


% Write out data to Excel files
xlswrite('Full Data Base.xls',enemy,'MANA Data Base','V2');

Enemy_Scenario=xlsread('Full Data Base.xls', 'MANA Data Base');

NearNaber = sortrows(Enemy_Scenario,22);

Output = NearNaber(1:k + 100,:);

for i = 1:k + 100
    % Calculate Euclidean Distance after finding Enemy Characteristics
first
    dist(i,1) = sqrt((Output(i,1)-Q(1,1))^2 + (Output(i,2)-Q(1,2))^2 +
(Output(i,3)-Q(1,3))^2 + (Output(i,4)-Q(1,4))^2 + ...
        (Output(i,5)-Q(1,5))^2 + (Output(i,6)-Q(1,6))^2 + (Output(i,7)-
Q(1,7))^2 + (Output(i,8)-Q(1,8))^2 + (Output(i,9)-Q(1,9))^2 ...
        + (Output(i,10)-Q(1,10))^2 + (Output(i,11)-Q(1,11))^2 +
(Output(i,12)-Q(1,12))^2 + (Output(i,13)-Q(1,13))^2 + (Output(i,14)-
Q(1,14))^2); %#ok<AGROW>
    % Calculate OEC with Fuzzy weights (C) from User Input
    OEC(i,1) = C(1,1)*(Output(i,1)/Q(1,1)) +
C(1,2)*(Output(i,2)/Q(1,2)) + C(1,3)*(Output(i,3)/Q(1,3)) +
C(1,4)*(Output(i,4)/Q(1,4)) + C(1,5)*(Output(i,5)/Q(1,5)) ...
```

```
        + C(1,6)*(Output(i,6)/Q(1,6)) + C(1,7)*(Output(i,7)/Q(1,7)) +
C(1,8)*(Output(i,8)/Q(1,8)) + C(1,9)*(Output(i,9)/Q(1,9)) +
C(1,10)*(Output(i,10)/Q(1,10)) ...
        + C(1,11)*(Output(i,11)/Q(1,11)) +
C(1,12)*(Output(i,12)/Q(1,12)) + C(1,13)*(Output(i,13)/Q(1,13))^2 +
C(1,14)*(Output(i,14)/Q(1,14)); %#ok<AGROW>
end


Request_from_user = Output(1:k + 100,:);
[Results, Header] = xlsread('Full Data Base.xls','MANA Data Base');
xlswrite('Query Results.xls',Header,'Query Results','A1');
xlswrite('Query Results.xls',Request_from_user,'Query Results','A2');
xlswrite('Query Results.xls',dist,'Query Results','S2');
xlswrite('Query Results.xls',OEC,'Query Results','U2');


Scenario2=xlsread('Query Results.xls', 'Query Results');
NearNaber2 = sortrows(Scenario2,19);

Output_Header = Header(1,15:22);
xlswrite('User Output.xls',Output_Header,'User Output','A1');
Output2=NearNaber2(1:k,15:22);
xlswrite('User Output.xls',Output2, 'User Output','A2');


% ---- Load Case #'s into the drop down menu list box in the GUI ----
function LoadCases = LoadCases(data, handles)

ListStr = []; %#ok<NASGU>

results_in = xlsread('User Output.xls');

ListStr = results_in(1:data.k,1);

set(handles.scenario_listbox, 'String', ListStr);

LoadCases = xlsread('Query Results.xls');

Case = data.CaseNum;

     for j = 1:data.k + 100
         if LoadCases(j,15) == Case
             RawCaseOut = LoadCases(j,1:14);
             CaseOut1 = RawCaseOut(1,1)/2880;
             CaseOut2 = RawCaseOut(1,2)/2880;
             CaseOut3 = RawCaseOut(1,3)/2;
             CaseOut4 = RawCaseOut(1,4);
             CaseOut5 = RawCaseOut(1,5);
             CaseOut6 = RawCaseOut(1,6)/2;
             CaseOut7 = RawCaseOut(1,7)/2;
             CaseOut8 = RawCaseOut(1,8);
             CaseOut9 = RawCaseOut(1,9);
             CaseOut10 = RawCaseOut(1,10)/2;
```

```
                CaseOut11 = RawCaseOut(1,11)*30;
                CaseOut12 = RawCaseOut(1,12);
                CaseOut13 = RawCaseOut(1,13)/2;
                CaseOut14 = RawCaseOut(1,14);
            end
        end

        set(handles.X1Var, 'String', CaseOut1)
        set(handles.X2Var, 'String', CaseOut2)
        set(handles.X3Var, 'String', CaseOut3)
        set(handles.X4Var, 'String', CaseOut4)
        set(handles.X5Var, 'String', CaseOut5)
        set(handles.X6Var, 'String', CaseOut6)
        set(handles.X7Var, 'String', CaseOut7)
        set(handles.X8Var, 'String', CaseOut8)
        set(handles.X9Var, 'String', CaseOut9)
        set(handles.X10Var, 'String', CaseOut10)
        set(handles.X11Var, 'String', CaseOut11)
        set(handles.X12Var, 'String', CaseOut12)
        set(handles.X13Var, 'String', CaseOut13)
        set(handles.X14Var, 'String', CaseOut14)


f1='C:\Documents and Settings\gtg730n\My Documents\MATLAB\Helo Flights
in Days - 3 sqds\Helo_-0-';

f2_=num2str(data.CaseNum);
f3='_pos_';
f4_=num2str(data.run);
f5='.csv';
file = strcat(f1,f2_,f3,f4_,f5);
RR=textread(file,'','delimiter',',','commentstyle','shell');

RRnew = sortrows(RR,2);
% Squad 1
S1 = sum(RRnew(:,2)==1);
s1 = 2*S1;
RR1 = RRnew(1:S1,:);

S1_supply(1,1) = 0;
S1_supply(1,2) = 100;

for t = 2:2:s1
    S1_supply(t,1) = (RR1(t/2,1));
    S1_supply(t,2) = 0;
    S1_supply(t+1,1) = (RR1(t/2,1));
    S1_supply(t+1,2) = 100;
end

S1_supply(s1 + 2,1) = 7;
S1_supply(s1 + 2,2) = 0;


% Squad 2
S2 = sum(RRnew(:,2)==2);
```

```
s2 = 2*S2;
start2 = S1 + 1;
end2 = S1 + S2;
RR2 = RRnew(start2:end2,:);

S2_supply(1,1) = 0;
S2_supply(1,2) = 100;

for t = 2:2:s2
    S2_supply(t,1) = (RR2(t/2,1));
    S2_supply(t,2) = 0;
    S2_supply(t+1,1) = (RR2(t/2,1));
    S2_supply(t+1,2) = 100;
end

S2_supply(s2 + 2,1) = 7;
S2_supply(s2 + 2,2) = 0;

% Squad 3
S3 = sum(RRnew(:,2)==3);
s3 = 2*S3;
start3 = S1 + S2 + 1;
end3 = S1 + S2 + S3;
RR3 = RRnew(start3:end3,:);

S3_supply(1,1) = 0;
S3_supply(1,2) = 100;

for t = 2:2:s3
    S3_supply(t,1) = (RR3(t/2,1));
    S3_supply(t,2) = 0;
    S3_supply(t+1,1) = (RR3(t/2,1));
    S3_supply(t+1,2) = 100;
end

S3_supply(s3 + 2,1) = 7;
S3_supply(s3 + 2,2) = 0;

%------Plot Squad Resupply History-------
s = data.squad;

 if s == 1;
     axes(handles.Resupply);
     x_axis = S1_supply(:,1);
     y_axis = S1_supply(:,2);
     plot(x_axis,y_axis)
     title 'Resupply Schedule History for Squad 1'
     xlabel('Time (Days)');
     ylabel({'Resupply Level'});
     grid on;

 elseif s == 2;
     axes(handles.Resupply);
     x_axis = S2_supply(:,1);
     y_axis = S2_supply(:,2);
```

```
    plot(x_axis,y_axis)
    title 'Resupply Schedule History for Squad 2'
    xlabel('Time (Days)');
    ylabel({'Resupply Level'});
    grid on;

elseif s == 3;
    axes(handles.Resupply);
    x_axis = S3_supply(:,1);
    y_axis = S3_supply(:,2);
    plot(x_axis,y_axis)
    title 'Resupply Schedule History for Squad 3'
    xlabel('Time (Days)');
    ylabel({'Resupply Level'});
    grid on;

elseif s == 0;
    axes(handles.Resupply);
    x1 = S1_supply(:,1);
    x2 = S2_supply(:,1);
    x3 = S3_supply(:,1);

    y1 = S1_supply(:,2);
    y2 = S2_supply(:,2);
    y3 = S3_supply(:,2);

    plot(x1,y1,x2,y2,x3,y3)
    title 'Resupply Schedule History for All Squads'
    legend('Sqd 1', 'Sqd 2', 'Sqd 3','Location','Best');
    xlabel('Time (Days)');
    ylabel({'Supply Level (%) '});
    grid on;

end
```

Appendix D
BACKGROUND RESEARCH

D 1. ANN Biological Architecture

Biological neural networks are made up of a vast quantity of neurons, which are the fundamental units of the nervous system. An individual neuron is a simple processing unit that receives and combines signals from other neurons through input paths called dendrites. Dendrites are bunched together into highly complex dendrite trees which are connected with the main body of the nerve cell. When a neuron reached a specified threshold level, the neuron fires and electrical signal along a single path called an axon. Neurons are connected to one another through the axon and adjacent dendrites via a connection called the synapse. A single neuron potentially may have between 1000 to 10,000 synapses and connected with thousands of neurons [61]. An example of a neural connection is shown below.

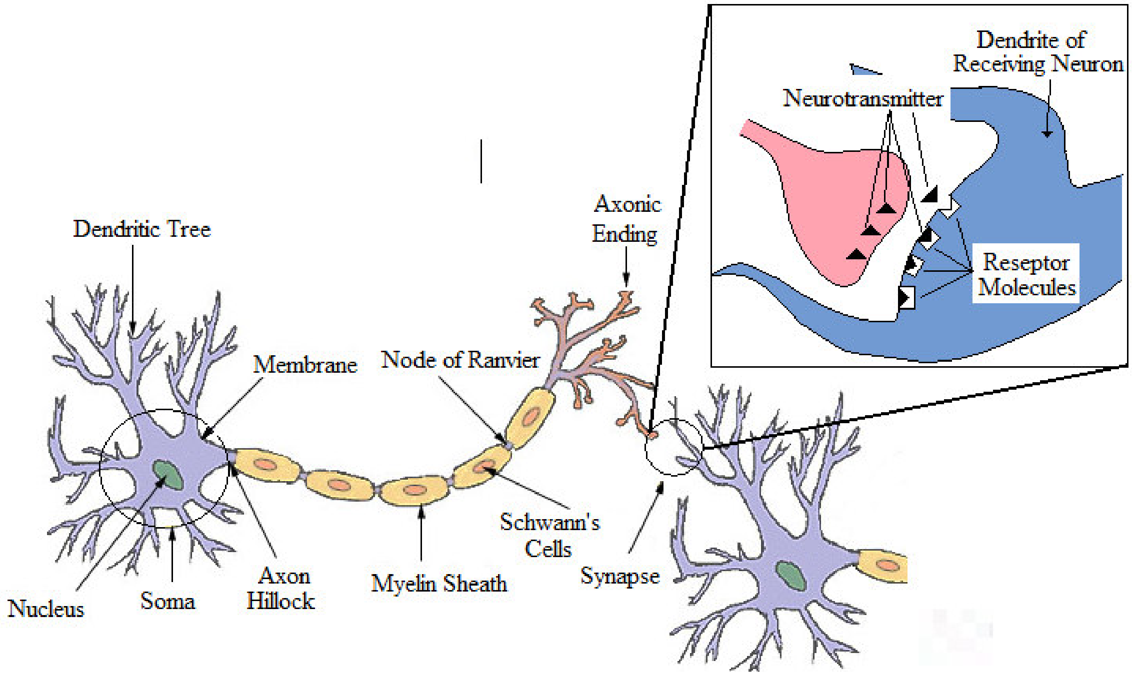


Figure 73: Representation of a Biological Neural Connection - Adapted from [44]

The signal transmitted via the synapse carries some electrical charge and as stated before in order to fire onto another synapse, must meet some threshold value [4]. This signal or charge is known as a neurotransmitter and will cause a postsynaptic potential to occur. This potential will either be excitatory (more positive) or inhibitory (more negative) [61]. Decoding at the synapse is accomplished by temporal summation and spatial summation as shown in the figure below [4].

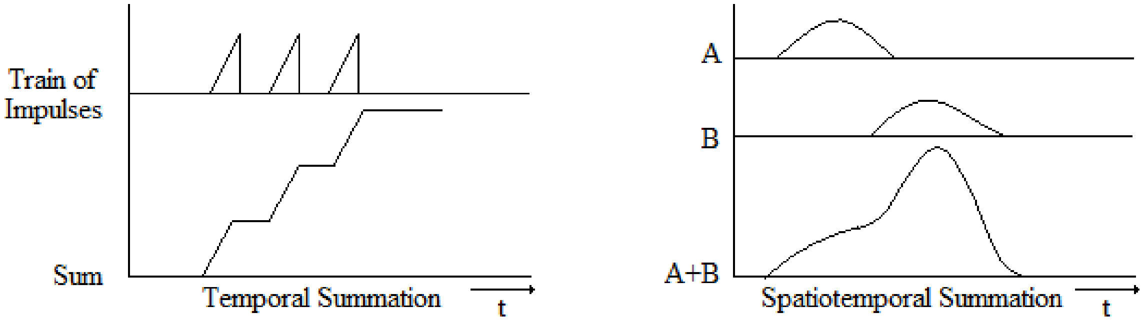


Figure 74: Summation at the Synapse (Adapted from [4])

The output signal can either be a single pulse or sequence of pulses at a particular rate and is transmitted to the axonic endings via the cells axon. Once this signal appears at the synapse, an electrical charge is generated. The magnitude of the charge depends on the incoming signal calculated, which in a way is a weighting factor associated with some input. This factor is generally not constant over a long period of time and is managed by the potassium ion flow which in turn changes the weight factors of the electrical signals send and processed in the neural pathways [61].

All neurons do not perform the exact same function or in the same way. The body is made up of many diverse neural pathways in which neurons may only activate a small subset of the genes within them. The diverse capability adds to the complexity of neural networks, but all neural networks exhibit certain properties, namely [61]:

- Parallel connections between many neurons
- Parallel connections have a feedback mechanism to other neurons and themselves
- Neurons may excite other neurons while at the same time inhibiting the functioning of others
- Some parts of the network may be existing and prewired whereas other my be evolutionary
- The output is not necessarily binary in nature
- They are asynchronous in operation
- They have a deliberate synchronizing mechanism that supports their vital functions
- They execute a program that is fully distributed and not sequentially executed
- Neural Networks do not have a central processor, they process things in a distributed fashion

D 2. Biologically Inspired Neural Networks

Some say there is a universal language that can be understood by all, but it is not language in the context of spoken word, it is mathematics. Many things man has found in nature inspire the development of systems or processes similar to or duplicating the abilities of natural processes. The body's neural processing power is no different and has been the subject of intense study by researchers and scientist as of late. The architecture of the man made neural network is composed of processing elements with characteristics such as inputs, weighting factors, activation, outputs and bias. Neural networks attempt to learn, recall, associate, and continuously compare new information with existing knowledge or data [61]. They also attempt to classify new information or inputs into existing categories or create new ones if necessary. With this in mind the process is then data driven and data is required to model the environment being investigated. Before the author gets ahead to the implementation and utilization phase, a basic understanding of the modeling and mathematics behind neural networks is required.

The processing element of a neural network is the artificial neuron (AN). Unfortunately since a biological neuron is quite complex, only a primitive version of an actual neuron has been replicated to date. A general AN has a set of N inputs, x_i, where the subscript i takes values from 1 to N and dictates the source of the input signal. Each input variable is weighted before reaching the primary processing element of the neuron by a weight factor, w_i; i.e. x_i is multiplies by w_i. In addition an AN may have a bias term w_0, a threshold value (θ) that must be achieved before the neuron will activate or fire, a function (f) that tends to be non linear in nature that acts on the produced signal (R_i), and finally an output. A mathematical representation of a neuron model is shown below:

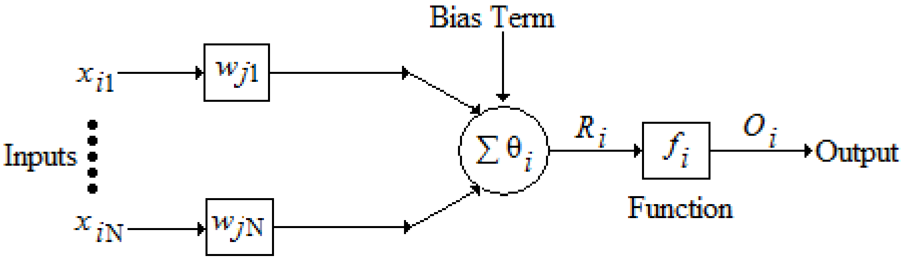


Figure 75: Mathematical Representation of Neuron [61]

The transfer function of a basic neuron model is described by the equation

$$O_i = f_i\left(\sum_{j=1}^{N} w_{ij} x_{ij}\right) \qquad \text{Equation 15}$$

and the neuron firing condition is

$$\sum_{j=1}^{N} w_{ij} x_{ij} \geq \theta_i \qquad \text{Equation 16}$$

with i signifying the neuron being looked at and j representing inputs for other neurons. The nonlinear function, f_i, helps to insure that the neurons response is bounded in the sense that when large or small activating stimuli occur the neuron is controllable by dampening or conditioning the signal. In the biological arena, conditioning occurs continuously and by all input sensors throughout the body or system being affected by a particular action signal. Several transfer functions can and are used to represent this activity with the logistic sigmoid or "squish" function being the most popular.

The squish function is very popular because it represents a nonlinear space, is monotonic, bounded, and has a simple derivative. Other functions may also be used such as the linear and step function. The figure below depicts a few popular functions available to the neural network designer.

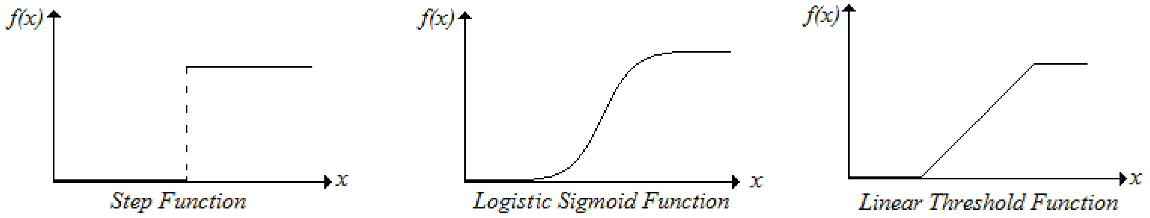


Figure 76: Different Types of Transfer Functions - Step, Sigmoid, and Linear [111]

The step function is non monotonic (with a discontinuity at the origin) and thus difficult to differentiate. It is however linear within its upper and lower bounds. All bounds are user defined for any option used to represent the neuron function component. The "squish" function is represented by the following:

$$output = \frac{1}{1 + e^{-\left(\sum_i w_i x_i + w_0\right)}} \qquad \text{Equation 17}$$

and always has an output between zero and one or negative one and one, depending on the designer setup. The linear threshold function allows for the summation of inputs and weighting without the discontinuity at the point of inflection like the step function.

Since the sigmoid function is the most common type of transfer function used an in depth overview will be given here. Before this can occur though, an understanding of the topology of ANN must be understood. ANN's consist of many neurons interconnected in certain ways to facilitate a desired response. These can range from fully connected to selected connections based on the network design. ANN's are made up of several components, the input layer, the hidden layer or layers, and the output layer. The input layer consists of the input variables that will be used to aid in the training and ultimately drive the outcomes of investigative activities in finding a solution to other problems. The hidden layer is where the regression analysis occurs and the input data is fit to a mathematical equation based on training data. There can be more than one hidden layer, but often times one is enough. It has been found that when a problem reaches some level of complexity, a neural network will often be more efficient with two smaller hidden layers instead of one large hidden layer [10]. The output layer is where the designer will get their responses to the specific problem being investigated. A diagram of a typical topology is shown below.

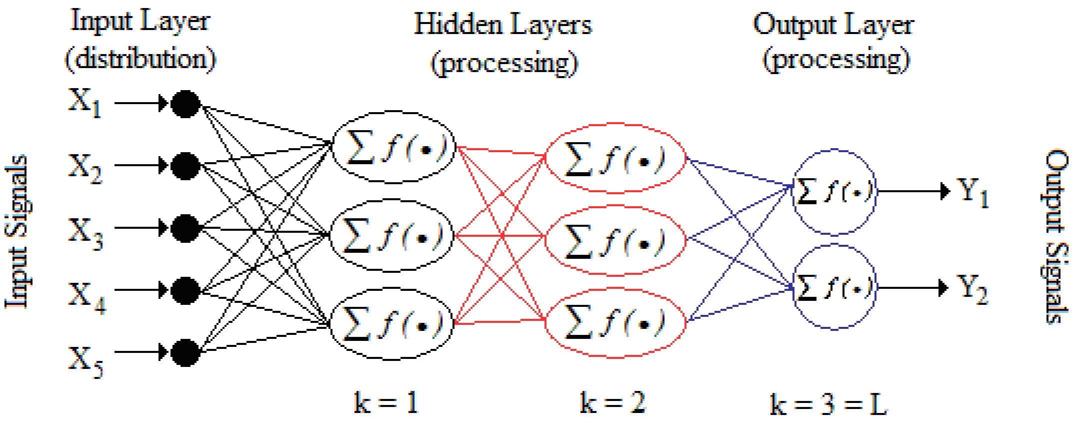


Figure 77: Multilayer Feed-forward Neural Network

Guidelines for determining what the best topology is, how many hidden layers there should be, or how many nodes in a layer are suitable are ambiguous. They are either heuristic or based on simulations derived from limited experiments [148]. The network in Figure 77 is *fully connected.* That is, every node in a particular layer receives inputs from every node in the preceding layer and projects its outputs to every node in the following layer [111]. For example in the network above, layer k = 2 receives inputs from every node in layer k = 1 and also outputs to every node in layer k =3 = L.

With a basic understanding of the general topology, an in depth look at the processing portion of the network can be described. The sigmoid function, as stated before, is the most common transfer function utilized because of its unique properties. It is used to calculate the value for each of the hidden nodes by applying the sigmoid function to a linear function of the input variables [56], as shown below.

$$H_j = S\left(a_j + \sum_{i=1}^{N}\left(b_{ij} X_i\right)\right) \qquad \text{Equation 18}$$

Where: a_j is the intercept term for the jth hidden node

b_{ij} is the coefficient for the ith design variable

X_i is the value of the ith design variable
H_j is the value of the jth hidden node
and N is the number of input variables

The response is calculated using a linear function that depends on the value of the hidden nodes with Equation 5. This equation is for the general case of k responses. Coefficients c_k and d_k are scale factors that represent the intercept and a multiplicative scalar on the interval [0,1] respectively.

$$R_k = c_k + d_k \left(e_k + \sum_{j=1}^{N_H} \left(f_{jk} H_j \right) \right) \qquad \text{Equation 19}$$

Where: c_k is the response scaling intercept term for the kth response
d_k is the response scaling coefficient for the kth response
e_k is the intercept term for the kth response
f_{jk} is the coefficient for the jth hidden node and kth response
H_j is the value of the jth hidden node (defined above)
and N_H is the number of hidden nodes

Equations 3 - 5 can be combined to create the following unified form of the neural network equation for function approximation.

$$R_k = c_k + d_k \left[e_k + \sum_{j=1}^{N_H} \left(f_{jk} \left(\frac{1}{1 + e^{-\left(a_j + \sum_{i=1}^{N} \left(b_{ij} X_i \right) \right)}} \right) \right) \right] \qquad \text{Equation 20}$$

Where: c_k is the response scaling intercept term for the kth response
d_k is the response scaling coefficient for the kth response
e_k is the intercept term for the kth response
f_{jk} is the coefficient for the jth hidden node and kth response
a_j is the intercept term for the jth hidden node

b_{ij} is the coefficient for the ith design variable

X_i is the value of the ith design variable

N is the number of input variables

and N_H is the number of hidden nodes

The six coefficients (*a, b, c, d, e,* and *f* above) are modified to fit the training set as well as possible, minimizing the sum square error (SSE) or mean square error (MSE) of the training set, depending on the training algorithm selected. That is, for *k* responses *R* and *N* design variables *X*, the neural network equation can be developed by selecting a number of hidden nodes N_H and determining values for the unknown scaling coefficients *a, b, c, d, e*, and *f* over the limits of summation illustrated in Equation 6. The process by which the scaling coefficients are determined is called training the neural network, and typically occurs through back-propagation of errors through the structure of neurons called a network [138]. The training process can be time intensive due to the large number of unknown coefficients.

Neural networks can also be utilized for classification problems. The output layer for the neural network consists of another layer of logistic sigmoid nodes instead of the linear transfer function used for function approximation [56]. The number of nodes in the output layer depends on the number of discrete responses (1 node per discrete response). Thus, each output node is a logistic sigmoid function of a summation of logistic sigmoid functions as shown below [56].

$$Output = S\left[c_k + \sum_{j=1}^{N_H} d_{jk}\left(S\left(a_j + \sum_{i=1}^{N}\left(b_{ij}X_i\right)\right)\right)\right] \qquad \text{Equation 21}$$

Expanded out the equation becomes the following,

$$Output = \frac{1}{1 + e^{-\left[c_k + \sum_{j=1}^{N_H} d_{jk}\left[\frac{1}{1+e^{-\left(a_j + \sum_{i=1}^{N} b_{ij} X_i\right)}}\right]\right]}} \qquad \text{Equation 22}$$

Where: a_j is the intercept term for the j$^{\text{th}}$ hidden node

b_{ij} is the coefficient for the i$^{\text{th}}$ design variable of the j$^{\text{th}}$ hidden node

X_i is the value of the i$^{\text{th}}$ design variable

N is the number of input variables

c_k is the intercept term for the k$^{\text{th}}$ response

d_{jk} is the coefficient for the j$^{\text{th}}$ hidden node and k$^{\text{th}}$ output node

and N_H is the number of hidden nodes

The resulting output nodes of the classification neural network are therefore between zero and one as opposed to the output of the function approximation neural network which can take on any value. The four coefficients (*a, b, c* and *d*) are modified to fit the training set by minimizing the sum square of errors. The training set generally is converted to zeros and one's before the training process is implemented.

The learning process for artificial neural networks is an area of intense research today. Biologist, engineers, behaviorist, etc are constantly trying to dissect how the human brain is able to process so much information and pass on that information in an efficient manner. Learning can generally be described as the process by which something is able to adapt itself to stimulus, and eventually after making the necessary adjustments is able to produce a desired response [61]. There are three methods of learning in neural networks; they are supervised training, unsupervised training, and reinforcement training.

Supervised training employs a "teacher" to aid in the training of the network by telling the neural net what the desired response should be to a particular stimulus. A l*earning algorithm* is a prescribed set of well-defined rules for learning of a neural network. There are many types of learning algorithms; the common goal of learning is the adjustment of connection weights. A block diagram below represents a typical flow of supervised learning.

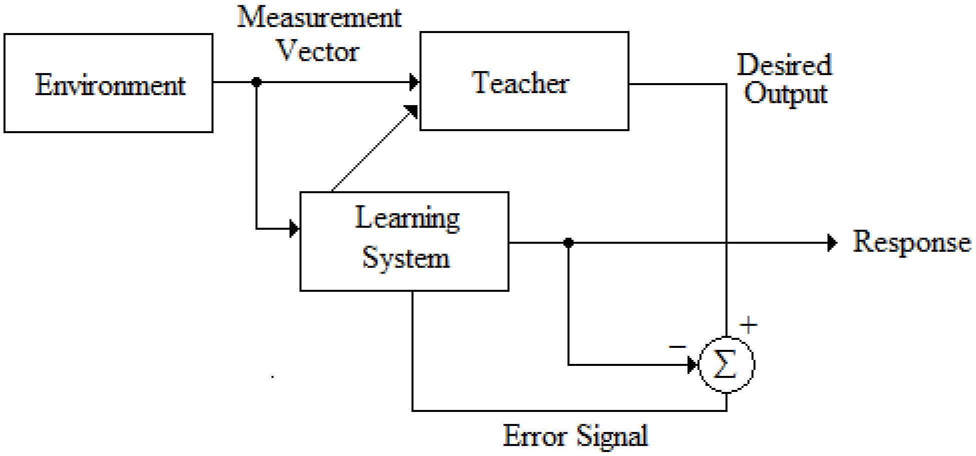


Figure 78: Supervised Learning Model for ANN [108]

Through this process the environment is exposed to the teacher and the learning environment through a vector of inputs. The teacher is able to, based on experience or other abilities, determine what the correct response to the inputs are. The learning environment also outputs what it thinks the "correct" answer is according to the inputs. A comparison is made between the two outputs and an error signal is generated. This error signal is used to adapt the weights and ultimately "trains" the network to attempt to generate an output that is within a prescribed error tolerance assigned to the network. The important principle for this method of training is that supervised learning requires an input and a corresponding desired output for this method to function [108].

Sometimes supervised training is not practical or possible, thus the method of unsupervised training can be utilized. This method is similar to the supervised method except there is no teacher. It is analogous to someone self teaching themselves through whatever method works best, often times trial and error or some other methodology to guide the network. This is represented by the block diagram below with the adaption rule block. Here the reader will also see the entire unsupervised method model.

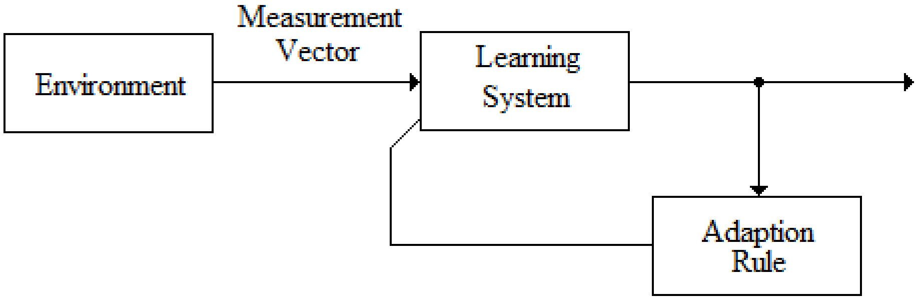


Figure 79: Unsupervised Learning Model for ANN [108]

There is no teacher involved in this process and the adaption rule is employed to adjust the weights of the learning system. As should be obvious, selection of the adaption rule or rules is a critical step and should be chosen and implemented by the designer after gaining a thorough understanding of the problem being investigated.

Reinforcement learning is where some form of data, x, is usually not given, but generated by an agent's interactions with the environment. At each point in time t, the agent performs an action y_t and the environment generates an observation x_t and an instantaneous cost c_t, according to some (usually unknown) dynamic.

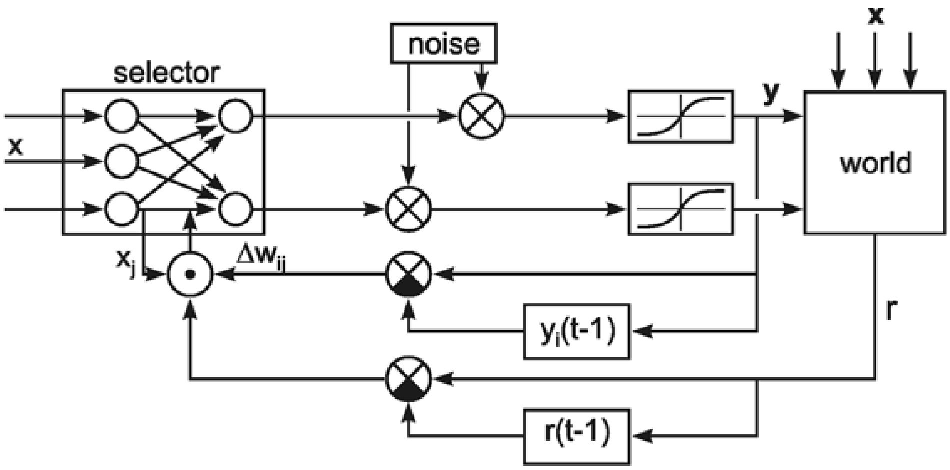


Figure 80: Reinforcement Learning Model for ANN [25]

The aim is to discover a *policy* for selecting actions that minimizes some measure of long-term cost, i.e. the expected cumulative cost. The environment's dynamics and the long-term cost for each policy are usually unknown, but can be estimated [5]. More formally, the environment can be modeled as a Markov Decision Process (MDP) with states s_1,…., s_n Є S and actions a_1,…, a_n Є A with the following probability distributions: the instantaneous cost distribution $P(c_t \mid s_t)$, the observation distribution $P(x_t \mid s_t)$ and the transition $P(s_{t+1} \mid s_t, a_t)$, while a policy is defined as conditional distribution over actions given the observations [5]. Taken together, these two define a Markov Chain (MC) where the aim is to discover a policy that minimizes the cost; i.e. the MC for which the cost is minimal.

In neural networks there are there are many types of methods used for training, that are used to describe the type of network being utilized. A general overview of the types can be seen in [5] or in Chapter 3 of [61]. I will briefly go over the feed forward and feed back or back propagation networks. A feed forward structure means that there are no connection loops that would allow any outputs to feed back to their inputs (indirectly) and change the output at a later time [111]. This type of network implements a static mapping that depends only on its present inputs and is completely independent of previous system states. Back propagation is, by far, the most popular and commonly used network. This term refers to two distinguishing points. First, it is a method that calculates the derivative of the network training error with respect to the weights by an application of the chain rule. Second, it describes a training algorithm, equivalent to gradient descent optimization, for using those derivatives to adjust the weights to minimize the error. This method and subsequent type of network is largely responsible for the renewed interest in neural networks in the mid 80's [111]. Its breakthrough was that it demonstrated that "layered networks of differentiable nonlinearities could perform useful, nontrivial calculations and that can offer (in some implementations) attractive features such as fast response, fault toleration, the ability to *learn* from examples, and some ability to generalize beyond the given training data" [111].

Unless the network is perfectly trained, the neural network outputs will vary from the desired outputs. The measurement error is the cost function, *e*, which can consist of many ways to measure. Typically used is the sum squared error (SSE) function shown below:

$$e = \frac{1}{2}\sum_{p}\sum_{i}\left(d_{pi} - y_{pi}\right)^2 \ , \qquad \text{Equation 23}$$

where *p* represent the patterns in the training set and, *i* represents the output nodes, and d_{pi} and y_{pi} are the expected output and the actual output, respectively, for the i^{th} output node on the p^{th} pattern [111].

Perhaps, the most primary significance of a neural network is the ability to learn the incoming information and to improve the performance of processing information. The term *learning* refers to many concepts by various viewpoints, and it is difficult to agree on a precise definition of the term. How it that we learn is still a question that remains unanswered. Johnson gives a very good overview in [56] of several types of optimization based training algorithms available in MATLAB.

D 3. Additional Review of Fuzzy Logic / Fuzzy Sets

Fundamental to fuzzy sets are the notion of union, intersection, and complement as defined similarly to classical set theory. The fuzzy union or disjunction (OR), fuzzy intersection or conjunction (AND), and fuzzy complement (NOT) are defined as classical operators for these functions respectively: OR = *max*, AND = *min*, and NOT = additive complement [78]. The union of two fuzzy sets A and B with respective membership functions $f_A(x)$ and $f_B(x)$ is a fuzzy set C, written as C = A ∪ B, whose membership function is related to those of A and B by:

$$f_c(x) = \mathrm{Max}\ [f_A(x), f_B(x)],\ \ x \in X \qquad \text{Equation 24}$$

or, in abbreviated form

$$fc = f_A \cup f_B. \qquad \text{Equation 25}$$

Note that union has the associative property, that is, A ∪ (B ∪ C) = (A ∪ B) ∪ C. The union of A and B is the smallest fuzzy set containing both A and B. More precisely, if D is any fuzzy set which contains both A and B, then it also contains the union of A and B. The notion of an intersection of fuzzy sets can be defined in an analogous manner. The *intersection* of two fuzzy sets A and B with respective membership functions $f_A(x)$ and

$f_B(x)$ is a fuzzy set C, written as C = A ∩ B, whose membership function is related to those of A and B by:

$$f_C(x) = \text{Min}\ [f_A(x), f_B(x)] \text{ where } x \in X, \qquad \text{Equation 26}$$

or, in abbreviated form,

$$f_C = f_A \cap f_B. \qquad \text{Equation 27}$$

As in the case of the union, it is easy to show that the intersection of A and B is the largest fuzzy set which is contained in both A and B. As in the case of ordinary sets, A and B are disjoint if A ∩ B is empty. Note that ∩, like ∪, has the associative property.

The complement of a fuzzy set A is denoted by A' and is defined by:

$$f_{A'}, = 1 - f_A. \qquad \text{Equation 28}$$

Most fuzzy problems have rules generated based on past experience to guide the solution to a correct output. This environment is called the fuzzy inference system (FIS) and consists of rule statements in the form of *IF (input fuzzy condition) THEN (output fuzzy assignment)* where the output designates a fuzzy value resulting from a causal relationship between the input and output fuzzy sets [105]. The input and output conditions are called the antecedent and consequent, respectively. Before generating these IF-THEN rule statements one should know all the input to output relationships in fuzzy terms. Simplification of these rules to the lowest level is also desired so as to reduce the level of complexity or possibility of going down the wrong path for a solution. If a rule does not fire, i.e. $f_A(x)=0$, the rule does not participate in the inference of the system as a whole.

The resulting output is useful in that it contains only qualitative information, most of the time though the output of a fuzzy system must be quantitative in the form of a crisp value. The process to convert the fuzzy output to a crisp one is called *defuzzification*. Several techniques have been developed to perform this process, but the three most used are the weighted average (self explanatory), mean of maxima (the maximum output is selected), and the most common method, center of areas (finds the output's center of mass) [61]. For mean of maxim method, given an output fuzzy set $Y = \mu_Y(a)$ defined in universe A of the variable a, the defuzzified output of several points (if they exist) for which the membership function is at its maximum is taken to be the mean of all the output values where the membership function is a maximum. It is represented by the equations below:

$$\begin{aligned} y_{\inf} &= \min(b \mid \mu_Y(b) = \max(\mu_Y(a))) \\ y_{\sup} &= \max(b \mid \mu_Y(b) = \max(\mu_Y(a))) \\ y_{MOM} &= \frac{y_{\inf} + y_{\sup}}{2} \end{aligned} \qquad \text{Equation 29}$$

The disadvantage of this method is that small changes in the system inputs can cause large movements in the defuzzified outputs [81]. For the center of areas method, the defuzzified output y_{COA} is given by the following equation [105]:

$$y_{COA} = \frac{\int_A a \cdot Y \; \partial v}{\int_A Y \; \partial v} \qquad \text{Equation 30}$$

If the output universe A is discrete the integral expressions are replaced and the resulting equation is shown below [140]:

$$y_{COA} = \frac{\sum_A a \cdot Y}{\sum_A Y} \qquad \text{Equation 31}$$

This method usually requires high computational resources and edge effects where a value that is very high when activated tends to draw the center of area towards itself [81].

Appendix E

E 1. Applications of Nonlinear Methods and Techniques

The following section investigates some of the design techniques in forecasting and prediction analysis in use today. In the realm of engineering the term forecasting is generally associated with the assessment or impact of new technologies or decisions made based on a proposed future plan. Generally these plans consist of where the company might be heading with a new design, its desired outcome, and of course a successful and profitable product with on time delivery. Much of the forecasting here is driven by expert opinion, but there is also a lot of data driven analysis that is accomplished by implementing some form of coding. Yet these codes must rely on predicted impacts of say technologies that may not exist yet. As in the case of aircraft design, this may consist of such things as a technology that provides a greater thrust to weight ratio, increased fuel efficiency, or reduced emissions.

In the business world forecasting deals with predicting customer needs as early as possible, sometimes even before the customer knows or decides they need the product. Much of this forecasting is data driven and is accomplished through surveys, past demand history, and sometimes expert opinion on how well a product, generally a new one, will perform in the market. Forecasting, particularly with neural networks, has been performed in the areas of image processing software, stock market prediction, engine management, currency risk management, automatic airplane landing systems, and power management on a regional and national scale. The list is endless in application areas.

In the manufacturing or operations management environment, supply chain forecasting and prediction on expected demand is accomplished with various methods depending on the available data and volatility of the demand environment. There are tools available that can allow analysts to make educated decision between the trade offs of "doing too much" or "doing too little" when faced with important ordering decision. For example consider a technology product that has a long lead time on components and it has a short shelf life because new technology makes it obsolete very quickly. Purchase too much and the company will likely be forced to sell its obsolete technology at a reduced price or take a complete loss. Buy too little, and like IBM experienced in the early 1990's with it's unexpected popular laptops, and the loss comes in the form of profits; in IBM's case it was estimated to be $100 million in demand [17]. One such tool to aid in this decision making process is the *Newsvendor Model*; it will be discussed later. Predicting anything in the future is a game of chance that will always be difficult. There are no concrete qualifications in terms of schooling or training that qualifies one to perform it well. Often time in the business world it takes someone to make a decision to put themselves or their company out on a limb and drive towards a goal to effect change and progress. Sometimes the result is success; other times a complete failure and possibly the dissolving of a once successful company. Like forecasting, prediction capabilities are also based on information to give the decision maker a better chance at being right.

E 2. Financial Markets Forecasting

If there is one area in the world that has an abundance of data it is the financial market. This market covers any thing a person could think of that has a financial impact.

Stock prices, interest rates, exchange rates, futures, bond rates, currency risk, etc. When it comes to money, much is documented and poured over by analyst looking to somehow decipher the evasive solution to performing well in the financial arena. A brief overview of this area is warranted because it has a similar environment to that of military logistics.

Military logistics can sometimes be very stable as is the interest and bond rates in the U.S. set by the Federal Reserve (Fed). Slight changes are made here and there depending on other events occurring globally, nationally, and sometimes locally. The Fed seeks to affect monetary policy, while the government affects fiscal policy; they have an inverse relationship to try to balance and keep the economy rolling along smoothly. Those involved with the futures market try to accurately predict the prices of such commodities as oil so as to buy low and sell high, thus making a profit. The military is the Presidents action arm in reacting and deterring dangerous action against the US and it allies. Providing the resources to the military so it is capable is very similar to the balancing act the Fed and the government perform and is the dilemma faced by the DOD; predicting where the next global instability might be so as to be ready to act and if possible prevent it before it occurs.

Then there are those in the financial markets that make their fortunes on large, short, and quick transaction. Quick decisions must be made on any and all available data; much of it is vague, incomplete, and raw in the sense of likely not very old. The outcome generally for those involved is to win big or to lose big, a very volatile environment. This is comparable to the environment military forces sometimes face during natural disasters or as in today’s world unforeseen acts by terrorist or radical extremist intent on inflicting harm to the U.S. A look at how exchange rates and currency risk can be managed will be conducted see how an environment such as this might be approached

In the past forecasting exchange rates was attempted via linear time series models. Over the course of time it was postulated and proven that this approach was incorrect because exchange rates, especially in today's every growing global financial markets, now exhibit non linear characteristics. Back propagation neural networks (BPN) have become a popular method for approaching the modeling of this environment. The authors of [69] applied this technique along with chaotic modeling with better predictive capabilities than with traditional statistical methods. The authors of [20] offer a modification to the BPN by incorporating a fuzzy interval architecture they call Fuzzy BPN. In this method the membership function is characterized by a Gaussian distribution in the form [20]:

$$f(x;c,\sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \qquad \text{Equation 32}$$

where c indicates the mean of the weekly exchange rate and σ represents the standard deviation of the weekly exchange rates. An illustration of this forecast environment is shown in the figure below.

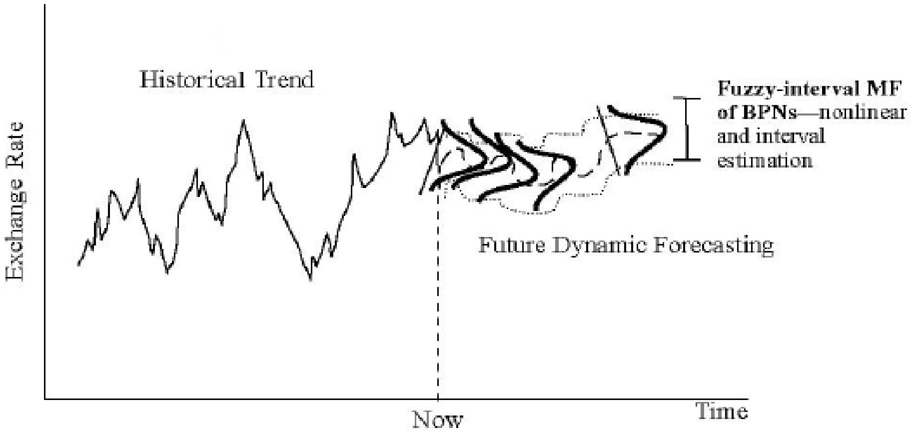


Figure 81: Fuzzy BPN Properties for Forecasting Exchange Rates [20]

To test their model the authors had at their disposal the bilateral exchange rate between the US Dollar and the New Taiwan Dollar for the time period between January of 1993 and October of 2006; this included 3425 observations. The ultimate goal was to predict the next following weeks 5 day trading exchange rate. The model is set up as a three layer feed forward network and is trained using a back propagation algorithm. There are up to two hidden layers allowed with the hidden nodes varying for one layer between 5, 15, 30, 50, and 100 and the second layer varying between 5, 15, and 30. Training is accomplished with 75% of the sample data and testing is performed with the remaining 25%. In the end the authors compare the mean square error of their proposed methodology, the original BPN, and another model called AR-GARSCH. The accuracy of their model is higher than the other two, but is still less than to be desired; only coming in at 83% with their training data and a high of 70% with their testing data. Despite its poor performance, overall the model performed well compared to others proposed.

E 3. Forecasting in Dynamic Environments-The Power Industry

This final area of exploration continues with the theme of being comparable to the dynamic environment of military logistics. The power industry is another entity where there is an abundance of data on usage and production rates. Load forecasting, an activity conducted on a daily and sometimes hourly basis, is an attempt by the power industry to plan ahead for times of peak demands and better allocate their resources to avoid disturbances for customers [7]. If this task were to be automated it would allow the Utilities, which are linked through a national grid to shift resources from area of low expected demand to those of higher expected demand. It would also allow companies to "sell" excess power producing capacity to other companies thus cutting down on everyone's over all cost because the necessity of turning on very expensive emergency generators available to meet fluxes in demand would be minimized. Most utility companies still use a human in the loop decision maker to forecast demand [7]. To conduct this predictive analysis factors such as the time of day, day of the week, the current weather conditions (humidity, wind speed, cloud cover, temperature) and whether it is a holiday or not are considered [7]. The authors of [7], [63], and [72] have proposed a variety of method utilizing ANN alone in the area of standard day forecasting and using NN with fuzzy inference systems where holidays are concerned. The reason for the two approaches is that for a standard regular day (weekday or weekend) there exist a vast amount of historical data and trends to draw on, thus theoretically providing the information needed to train and test a neural network system. When holidays are concerned there are limitations in knowledge with anomalous load condition, thus it is difficult to obtain a sufficient data set for the NN to learn those peculiar load profiles.

Moreover, ANN alone cannot deal with the qualitative knowledge and experience of the human forecasters who provide accurate forecasting results for these "special" days [72].

The first methodology for regular day forecasting has an architecture that is described as follows. There are 24 tightly coupled NN, one for every hour of the day. Each network consists of 12 inputs (4 weather parameters for three cities being analyzed), two to five hidden nodes, and one output. Each neural network is fully connected with its two closest neighbors, thus each node receives an additional input from the previous and the next hour weather conditions of the training data. This in effect encode two overlapping three hour weather trends, making the output unit "aware" of a five hour weather trend around the hour it is analyzing. This architecture has the effect of reducing the number of connections to 1776 instead of 15024 that would occur in a fully connected network; a reduction in connection weights by a factor of 10 [7]. Three separate networks were created with differing amounts of information for the time period of March 1985. The table below summarizes this information and the results.

Table 22: Results from Load Forecasting for ANN alone

Neural Network	Data Used	Convergence Reached (# of epochs)	Accuracy within 5% of target value
Net 1	All inputs minus day of the week	10500	79%
Net 2	All inputs	9800	94 %
Net 3	Temperature only	Reached stopping criteria 50000	85%

A review of the table above shows some interesting results. One would have expected the rank to be Net 2, Net 1, and Net 3 in order of accuracy. The authors of [7] found that this was likely because there were only three weekends worth of data in the training data available and that although Net 1 did perform better, its forecast were very erratic at times and showed sharp fluctuations. Of note, irregularities in the outputs that seemed odd actually pointed the designers to data that had been errantly entered into the training data file as a result of human error.

The most interesting methodology application for the power industry segment is conducted in [63]. The goal of this architecture is to predict holiday usage where demand is lower due to the fact that many businesses close for the day. The FIS is created by using a variable called load difference which measures the difference between a typical day and past measurements of a holiday's usage data. The variable uses are typical load difference maximum (TLD) and minimum, variance of load difference (VLD) maximum and minimum and load difference (LD) maximum and minimum. Here a fuzzy inference system is generated with fuzzy input variables and triangular membership functions resulting in 121 fuzzy rules. In this process scaled load curves are utilized to aid in the training process of the fuzzy neural network. Every ANN was trained using an error back propagation algorithm with an adaptive learning rate and momentum [138]. The training procedure was terminated if the maximum of the daily averaged error, as calculated by Equation 24 was less than 1%. An attempt to avoid over fitting or under fitting was sought by varying the stopping criteria between 0.5 % and 2 % [63]. Results from this architecture were very promising in that the nonlinear relationships from special holidays were substantially captured through the use of a fuzzy inference system and neural network. A look at the Korean Thanksgiving holiday showed an average forecast error of 2.64 % with a maximum of 6.40% for the proposed methodology versus two other methods which showed average errors in forecasting of 3.97% and 8.13% and maximum errors of 11.54% and 13.46%, respectively.

Appendix F
OPLOGPLN Reports

This appendix contains all the pertinent reports that were generated by the author's execution of the Operations Logistics Planner Tool that was acquired from the Combined Armed Service Command located in Ft. Lee, Virginia. This tool is currently what is used to plan for logistics requirements at a top level based on aggregated and expected force size as well as a variety of mission characteristics. The reports in this appendix consist of the results from OPLOGPLN that encapsulate the forecasts for Class I (food), Class II (general supplies), Class III (fuel), Class IV (construction materials), Class V (ammunition), Class VI (personal items), Class VIII (medical materials), Water, and the unit Equipment List. The last report verifies to the user what pieces of equipment, weapons, or other end items that are either part of the military unit (as specified by the TO&E) or have been added to make up the (MTO&E). The reports provide a breakdown of each class by each of the three phases of the operation specified by the author. These phases consist of the deployment phase, operational or mission phase, and then the recovery or return to home base phase. Each of these phases has a breakdown in quantities of specific items, the total weight, and in some instances the number of cases or pallets that are projected based on the packaging characteristics of the item in question.

F 1. Class I

CLASS I

OPLOG Planner
Population Based Logistics Estimate

Phase Summary Report

Totals for Operation:

	Quantity	Lbs	Cases	Pallets
MEALS:				
MRE	1,056	1,932.5	88.0	1.8

Class I Totals for Operation: 1,932.5 **Lbs** 1.8 **Pallets**

UNCLASSIFIED

Class I: Phase Summary Report

Operation Name: MyOrder Length of Phase: 1 days

Totals for Phase: 1 **Insertion**

	Quantity	Lbs	Cases	Pallets
MEALS:				
MRE	132	241.6	11.0	0.2

Class I Totals for Phase: 1 241.6 **Lbs** 0.2 **Pallets**

UNCLASSIFIED

Class I: Phase Summary Report

Operation Name: MyOrder Length of Phase: 6 days

Totals for Phase: 2 **Defend Border**

	Quantity	Lbs	Cases	Pallets
MEALS:				
MRE	792	1,449.4	66.0	1.4

Class I Totals for Phase: 2 1,449.4 **Lbs** 1.4 **Pallets**

UNCLASSIFIED

Class I: Phase Summary Report

Operation Name: MyOrder Length of Phase: 1 days

Totals for Phase: 3 **Extraction**

	Quantity	Lbs	Cases	Pallets
MEALS:				
MRE	132	241.6	11.0	0.2

Class I Totals for Phase: 3 241.6 **Lbs** 0.2 **Pallets**

F 2. Class II

UNCLASSIFIED

CLASS II

OPLOG Planner
Population Based Logistics Estimate

Phase Summary Report

Totals for Operation:

	Lbs	Short Tons	Pallets
Class II Totals:	728.2	0.4	1.8

Phase: 1
Task Organization: DO

Lbs	Short Tons	Pallets
158.4	0.1	0.4

Totals for Phase: 1 **Insertion**
Length of Phase: 1 days

Lbs	Short Tons	Pallets
158.4	**0.1**	**0.4**

Phase: 2
Task Organization: DO

Lbs	Short Tons	Pallets
488.4	0.2	1.2

Totals for Phase: 2 **Defend Border**
Length of Phase: 6 days

Lbs	Short Tons	Pallets
488.4	**0.2**	**1.2**

Phase: 3
Task Organization: DO

Lbs	Short Tons	Pallets
81.4	0.0	0.2

UNCLASSIFIED

Totals for Phase: 3 **Extraction**
Length of Phase: 1 days

Lbs	Short Tons	Pallets
81.4	**0.0**	**0.2**

F 3. Class III (Bulk)

UNCLASSIFIED

CLASS III (B)

OPLOG Planner
Equipment Based Logistics Estimate

Phase Report

Total for Operation

	POL Type	Gallons	Tanker Loads (2500 gal equivalents)
Class III (B)	**JP8:**	2,541.0	1.06
	Total:	2,541.0	1.06

Phase: 1
Task Organization: DO

	POL Type	Gallons	Tanker Loads (2500 gal equivalents)
Class III (B)	**JP8:**	317.6	0.13

Total for Phase: 1
Length of Phase: 1 days

	POL Type	Gallons	Tanker Loads (2500 gal equivalents)
Class III (B)	**JP8:**	317.6	0.13

UNCLASSIFIED

Phase: 2
Task Organization: DO

	POL Type	Gallons	Tanker Loads (2500 gal equivalents)
Class III (B)	**JP8:**	1,905.7	0.79

Total for Phase: 2
Length of Phase: 6 days

	POL Type	Gallons	Tanker Loads (2500 gal equivalents)
Class III (B)	**JP8:**	1,905.7	0.79

Phase: 3
Task Organization: DO

	POL Type	Gallons	Tanker Loads (2500 gal equivalents)
Class III (B)	**JP8:**	317.6	0.13

Total for Phase: 3
Length of Phase: 1 days

	POL Type	Gallons	Tanker Loads (2500 gal equivalents)
Class III (B)	**JP8:**	317.6	0.13
	Total:	317.6	0.13

NOTE: "Tanker Loads" assumes that only 2400 gals and be transported in a 2500 gal Tanker.

F 4. Class III (POL)

UNCLASSIFIED

CLASS III (P)

OPLOG Planner
Equipment Based Logistics Estimate

Phase Report

Totals for Operation:

	Lbs	Short Tons	Pallets
Class III(P) Totals:	146.0	0.1	0.2

Totals for Phase: 1 **Insertion**

Length of Phase: 1 days	Lbs	Short Tons	Pallets
	18.2	0.0	0.0

Totals for Phase: 2 **Defend Border**

Length of Phase: 6 days	Lbs	Short Tons	Pallets
	109.5	0.1	0.1

Totals for Phase: 3 **Extraction**

Length of Phase: 1 days	Lbs	Short Tons	Pallets
	18.2	0.0	0.0

UNCLASSIFIED

F 5. Class IV

UNCLASSIFIED

CLASS IV

OPLOG Planner

Population/Task Based Logistics Estimate

Phase Report

Totals for Operation

	Lbs	Short Tons	Pallets
Barrier	823.7	0.4	1.6
Construction	714.6	0.4	1.4
Class IV - Total	**1,538.2**	**0.8**	**2.9**

Totals for Phase: 1 — Insertion

Length of Phase: 1 days	Lbs	Short Tons	Pallets
Barrier	103.0	0.1	0.2
Construction	89.3	0.0	0.2
Class IV - Total	**192.3**	**0.1**	**0.4**

UNCLASSIFIED

Totals for Phase: 2 — Defend Border

Length of Phase: 6 days	Lbs	Short Tons	Pallets
Barrier	617.8	0.3	1.2
Construction	535.9	0.3	1.0
Class IV - Total	**1,153.7**	**0.6**	**2.2**

Totals for Phase: 3 — Extraction

Length of Phase: 1 days	Lbs	Short Tons	Pallets
Barrier	103.0	0.1	0.2
Construction	89.3	0.0	0.2
Class IV - Total	**192.3**	**0.1**	**0.4**

UNCLASSIFIED

F 6. Class V

CLASS V

OPLOG Planner
System Based Logistics Estimate

Unit/Phase LIN level Report

Totals for Operation	LBS	Short Tons	Cubic Ft.	Pallets
	809.43	0.40	15.63	0.39

Phase: 1 Length of Phase: 1 days

DODIC	DODIC Description	Qty Used	LBS	Short Tons	Cubic Ft.	Pallets
A059	CTG 5.56 MM BALL, M855 10RD CLIP	66	2.61	0.00	0.04	0.00
A063	CTG 5.56 MM TR M856 SINGLE RD	16	0.67	0.00	0.01	0.00
A064	CTG 5.56 MM 4 BALL, 1 TRACER LKD	57	3.34	0.00	0.07	0.00
A131	CTG 7.62 MM 4 BALL M80/1 TR M62	19	1.99	0.00	0.03	0.00
A363	CTG 9 MM BALL NATO, M882	1	0.04	0.00	0.00	0.00
A518	CTG .50 CAL 4 SLAP M903 / 1 SLAP-	25	9.61	0.00	0.11	0.00
A557	CTG .50 CAL 4 BALL + 1 TR	14	5.30	0.00	0.06	0.00
A576	CTG 50 CAL 4 API M8/1 API-T M20	56	20.94	0.01	0.32	0.01
A607	CTG .50 CAL 4 API MK211 / 1 API-T	14	5.52	0.00	0.06	0.00
AA02	CTG 5.56 MM 4 AP M995/1 TR M865	53	2.44	0.00	0.04	0.00
AA04	CTG 7.62 MM 4 API M933 / 1 TR M62	10	0.90	0.00	0.01	0.00
B504	CTG 40 MM GRN STAR PARA M661	0	0.30	0.00	0.01	0.00
B505	CTG 40 MM RED STAR PARA M662	0	0.30	0.00	0.01	0.00
B506	CTG 40 MM RED SMK M713	0	0.15	0.00	0.00	0.00
B508	CTG 40 MM GRN SMK M715	0	0.15	0.00	0.00	0.00
B509	CTG 40 MM YLW SMK M716	0	0.15	0.00	0.00	0.00
B534	CTG 40 MM MP M576	0	0.21	0.00	0.01	0.00
B535	CTG 40 MM ILLUM WHT STAR	0	0.45	0.00	0.01	0.00
B536	CTG 40 MM ILLUM WHT STAR	0	0.30	0.00	0.01	0.00
B542	CTG 40 MM HEDP M430 LINKED	31	38.58	0.02	1.02	0.03
B546	CTG 40 MM HEDP M433/ PA120	2	1.26	0.00	0.04	0.00
BA03	CTG 40 MM INFRARED	1	0.60	0.00	0.00	0.00
BA11	CTG 40 MM CANNISTER M1001	4	5.38	0.00	0.09	0.00
Totals for Phase: 1			101	0.05	1.95	0.05

Phase: 2 Length of Phase: 6 days

DODIC	DODIC Description	Qty Used	LBS	Short Tons	Cubic Ft.	Pallets
A059	CTG 5.56 MM BALL, M855 10RD CLIP	393	15.68	0.01	0.23	0.01
A063	CTG 5.56 MM TR M856 SINGLE RD	98	4.02	0.00	0.06	0.00
A064	CTG 5.56 MM 4 BALL, 1 TRACER LKD	342	20.07	0.01	0.45	0.01

A131	CTG 7.62 MM 4 BALL M80/1 TR M62	116	11.95	0.01	0.20	0.01
A363	CTG 9 MM BALL NATO, M882	6	0.23	0.00	0.00	0.00
A518	CTG .50 CAL 4 SLAP M903 / 1 SLAP-	150	57.66	0.03	0.67	0.02
A557	CTG .50 CAL 4 BALL + 1 TR	83	31.80	0.02	0.37	0.01
A576	CTG 50 CAL 4 API M8/1 API-T M20	335	125.64	0.06	1.90	0.05
A607	CTG .50 CAL 4 API MK211 / 1 API-T	86	33.11	0.02	0.39	0.01
AA02	CTG 5.56 MM 4 AP M995/1 TR M865	317	14.64	0.01	0.24	0.01
AA04	CTG 7.62 MM 4 API M933 / 1 TR M62	58	5.42	0.00	0.07	0.00
B504	CTG 40 MM GRN STAR PARA M661	2	1.78	0.00	0.04	0.00
B505	CTG 40 MM RED STAR PARA M662	2	1.78	0.00	0.04	0.00
B506	CTG 40 MM RED SMK M713	1	0.90	0.00	0.02	0.00
B508	CTG 40 MM GRN SMK M715	1	0.90	0.00	0.02	0.00
B509	CTG 40 MM YLW SMK M716	1	0.90	0.00	0.02	0.00
B534	CTG 40 MM MP M576	2	1.26	0.00	0.03	0.00
B535	CTG 40 MM ILLUM WHT STAR	3	2.68	0.00	0.06	0.00
B536	CTG 40 MM ILLUM WHT STAR	2	1.78	0.00	0.04	0.00
B542	CTG 40 MM HEDP M430 LINKED	185	231.46	0.12	6.12	0.15
B546	CTG 40 MM HEDP M433/ PA120	10	7.54	0.00	0.21	0.01
BA03	CTG 40 MM INFRARED	3	3.57	0.00	0.00	0.00
BA11	CTG 40 MM CANNISTER M1001	25	32.31	0.02	0.54	0.01
Totals for Phase: 2			607	0.30	11.72	0.29

Phase: 3 Length of Phase: 1 days

DODIC	DODIC Description	Qty Used	LBS	Short Tons	Cubic Ft.	Pallets
A059	CTG 5.56 MM BALL, M855 10RD CLIP	66	2.61	0.00	0.04	0.00
A063	CTG 5.56 MM TR M856 SINGLE RD	16	0.67	0.00	0.01	0.00
A064	CTG 5.56 MM 4 BALL, 1 TRACER LKD	57	3.34	0.00	0.07	0.00
A131	CTG 7.62 MM 4 BALL M80/1 TR M62	19	1.99	0.00	0.03	0.00
A363	CTG 9 MM BALL NATO, M882	1	0.04	0.00	0.00	0.00
A518	CTG .50 CAL 4 SLAP M903 / 1 SLAP-	25	9.61	0.00	0.11	0.00
A557	CTG .50 CAL 4 BALL + 1 TR	14	5.30	0.00	0.06	0.00
A576	CTG 50 CAL 4 API M8/1 API-T M20	56	20.94	0.01	0.32	0.01
A607	CTG .50 CAL 4 API MK211 / 1 API-T	14	5.52	0.00	0.06	0.00
AA02	CTG 5.56 MM 4 AP M995/1 TR M865	53	2.44	0.00	0.04	0.00
AA04	CTG 7.62 MM 4 API M933 / 1 TR M62	10	0.90	0.00	0.01	0.00
B504	CTG 40 MM GRN STAR PARA M661	0	0.30	0.00	0.01	0.00
B505	CTG 40 MM RED STAR PARA M662	0	0.30	0.00	0.01	0.00
B506	CTG 40 MM RED SMK M713	0	0.15	0.00	0.00	0.00
B508	CTG 40 MM GRN SMK M715	0	0.15	0.00	0.00	0.00
B509	CTG 40 MM YLW SMK M716	0	0.15	0.00	0.00	0.00
B534	CTG 40 MM MP M576	0	0.21	0.00	0.01	0.00
B535	CTG 40 MM ILLUM WHT STAR	0	0.45	0.00	0.01	0.00
B536	CTG 40 MM ILLUM WHT STAR	0	0.30	0.00	0.01	0.00
B542	CTG 40 MM HEDP M430 LINKED	31	38.58	0.02	1.02	0.03
B546	CTG 40 MM HEDP M433/ PA120	2	1.26	0.00	0.04	0.00
BA03	CTG 40 MM INFRARED	1	0.60	0.00	0.00	0.00
BA11	CTG 40 MM CANNISTER M1001	4	5.38	0.00	0.09	0.00
Totals for Phase: 3			101	0.05	1.95	0.05

F 7. Class VI

UNCLASSIFIED

CLASS VI

OPLOG Planner
Population Based Logistics Estimate

Phase Report

Totals for Operation:

	LBS	Short Tons	Pallets	40ft Container
Class VI	197.12	0.10	0.40	0.01

Totals for Phase: 1 Insertion

Length of Phase: 1 days

	LBS	Short Tons	Pallets	40ft Container
Class VI	24.64	0.01	0.05	0.00

Totals for Phase: 2 Defend Border

Length of Phase: 6 days

	LBS	Short Tons	Pallets	40ft Container
Class VI	147.84	0.07	0.30	0.01

Totals for Phase: 3 Extraction

Length of Phase: 1 days

	LBS	Short Tons	Pallets	40ft Container
Class VI	24.64	0.01	0.05	0.00

UNCLASSIFIED

F 8. Class VIII

UNCLASSIFIED

CLASS VIII

OPLOG Planner
Casualty/Population Based Logistics Estimate

Phase Report

Totals for Operation:

	LBS	Short	Pallets
Class VIII	52.8	0.03	0.21

Totals for Phase: 1 Insertion
Length of Phase: 1 days

LBS	Short Tons	Pallets
6.6	0.00	0.03

Totals for Phase: 2 Defend Border
Length of Phase: 6 days

LBS	Short Tons	Pallets
39.6	0.02	0.16

Totals for Phase: 3 Extraction
Length of Phase: 1 days

LBS	Short Tons	Pallets
6.6	0.00	0.03

UNCLASSIFIED

F 9. Water

UNCLASSIFIED

WATER

OPLOG Planner
Population Based Logistics Estimate

Phase Report

Total for Operation:

	Potable		Non-Potable
	Bulk	**Bottled**	
Unit Level Gals:	**1,705.4**	**0.0**	
SubTotal Gals:	**1,705.4**	**0.0**	**0.0**
Total Gals:	**1,705.4**		**0.0**

Totals for Phase: 1 Insertion
Length of Phase: 1 days

	Potable		Non-Potable
	Bulk	**Bottled**	
Unit Level Gals:	**213.2**	**0.0**	
SubTotal Gals:	**213.2**	**0.0**	**0.0**
Total Gals:	**213.2**		**0.0**

Totals for Phase: 2 Defend Border
Length of Phase: 6 days

	Potable		Non-Potable
	Bulk	**Bottled**	
Unit Level Gals:	**1,279.0**	**0.0**	
UEx/UEy Level	**0.0**		**0.0**
SubTotal Gals:	**1,279.0**	**0.0**	**0.0**
Total Gals:	**1,279.0**		**0.0**

Totals for Phase: 3 Extraction
Length of Phase: 1 days

	Potable		Non-Potable
	Bulk	**Bottled**	
Unit Level Gals:	**213.2**	**0.0**	
SubTotal Gals:	**213.2**	**0.0**	**0.0**
Total Gals:	**213.2**		**0.0**

UNCLASSIFIED

F 10. Equipment List

UNCLASSIFIED

Equipment List

OPLOG Planner

Unit level Phase Report

Phase: 1, 2, 3 Phase: Insertion, Defend Border, Extraction

Joint Phase: Dominate Military Operation: Offensive
Task Org: DO Task Org: TO1

	DO Platoon: CUSTOM, Quantity: 1	Unit	Individual
C0693	CARBINE 5.56 MILLIMETER: M4A1	26	26
C7768	COMPUTER: FIRE CONTROL AN/PSG-8	1	1
G97730	GUN LAYING AND POSITIONING SYS:	3	3
L6901	LAUNCHER GRENADE: 40MM, M203E2	9	9
L9197	MACHINE GUN CALIBER .50: HB	4	4
M09009	MACHINE GUN 5.56 MILLIMETER: M249	9	9
M29999	MEDICAL EQUIPMENT SET SPECIAL	1	1
M60256	MOUNT TRIPOD MACH GUN 7.62MM:	1	1
M92362	MACHINE GUN GRENADE 40MM: MK19	4	4
M92841	MACHINE GUN: 7.62MM M240B	3	3
N0445	NIGHT VISION GOGGLE: AN/PVS-5	10	10
N0459	NIGHT VISION SIGHT CREW SERVED	3	3
N0473	NIGHT VISION SIGHT INDIVIDUAL	8	8
N0548	NIGHT VISION GOGGLE: AN/PVS-7B	1	1
P9815	PISTOL 9MM AUTOMATIC: M9	2	2
R2979	RECEIVER SET RADIO: AN/PRR-9	44	44
T6156	TRUCK UTILITY: CARGO/TROOP	11	11
Z0004	SATELLITE COMMUNICATIONS EARTH	5	5
Z0044	LASER MARKER: AN/PEQ-1B	11	11
Z0087	HF RADIO SET: AN/PRC-150C MANPACK	1	1
Z0087	MANPACK TACSAT AN/PRC-117F	5	5
Z0090	ILLUMINATOR, INTEGRATED, SMALL	11	11
Z0230	MOUNT TRIPOD F/M240B MACHINE GUN	2	2
Z2833	FREQUENCY HOPPING MULTIPLEXER:	11	11
Z9996	RADIO SET: AN/PRC-148 (V) 2 (C)	11	11

UNCLASSIFIED

REFERENCES

[1] Aamodt, A. and Plaza, E. "Case based Reasoning: Foundational Issues, Methodological Variations, and System Approaches." AI – Communications, v 7, n 1, 1994, p. 39-59.

[2] Adelman, L. Evaluating Decision Support and Expert Systems. New York, John Wiley, 1992.

[3] AH1-Z Viper. Wikipedia: The Free Online Encyclopedia. Accessed November 4, 2008. http://en.wikipedia.org/wiki/AH-1Z_Viper.

[4] Alkon, D.L. and Rasmussen, H. "A Spatial-Temporal Model of Cell Activation," Science, Vol. 239, No. 4843, 1988. p. 998-1005.

[5] Artificial Neural Network: Wikipedia: The Free Online Encyclopedia. Accessed November 3, 2007. http://en.wikipedia.org/wiki/Artificial_neural_network.

[6] Autoregressive Integrated Moving Average from Wikipedia: The Free Online Encyclopedia. Accessed January 8, 2008. http://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average.

[7] Bacha, Hamid and Meyer, Walter. "A Neural Network architecture for Load Forecasting." International Joint Conference on Neural Networks, Vol. 2, June 7-11, 1992. p. 442 - 447.

[8] Bain, Matthew D. "Supporting a Marine Corps Distributed Operations Platoon: A Quantitative Analysis." MS Dissertation: Naval Post Graduate School, Monterey CA. September 2005.

[9] Baker III, F. W. "International War Game Explores Seabasing Concept." American Forces Press Service. March 6, 2008.

[10] Barron, A.R. "A Comment on Neural Networks: A Review from a Statistical Perspective". Statistical Science, 9 (1), 1994. p. 33–35.

[11] Ben-Haim, Y. Information Gap – Decision Theory – Decision Under Severe Uncertainty. Academic Press: New York, 2001.

[12] Berg, B. Markov Chain Monte Carlo Simulations and Their Statistical Analysis. New Jersey: World Scientific, 2004.

[13] Biltgen, P. "A Methodology for Capability Based Technology Evaluation for System of Systems." PhD Thesis, Georgia Institute of Technology, May 2007.

[14] Bitinas, E, Henscheid, Z., and Truong, L. "Pythagoras: A New Agent-Bases Simulation System." Technology Review Journal. Spring/Summer 2003.

[15] Bottani, Eleonora and Rizzi, Antonio. "Strategic Management of Logistics Service: A Fuzzy QFD Approach." International Journal of Production Economics, Issue 103, 2006.

[16] Box, G. E. P. and Draper, N. R. Empirical Model Building and Response Surfaces. New York: John Wiley and Son, 1987.

[17] Cachon, G, and Terwiesch, C. Matching Supply with Demand. New York: McGraw Hill, 2006.

[18] Calculating Demand Forecast Accuracy from Wikipedia: The Free Online Encyclopedia. Accessed Jan 17, 2008. http://en.wikipedia.org/wiki/Calculating_Demand_Forecast_Accuracy.

[19] Campos, F. and Cavalcante S. "An Extended Approach for Dempster-Shafer Theory." IEEE Explore, 2003.

[20] Carlsson, C., Fedrizzi, M., Fuller, R. Fuzzy Logic in Management. Massachusetts: Kluwer Academic Publishers, 2004.

[21] CH53E Super Stallion. Wikipedia: The Free Online Encyclopedia. Accessed November 4, 2008. http://en.wikipedia.org/wiki/CH-53E_Super_Stallion.

[22] Chavers, T. OPLOG Planner Briefing, United States Army Combined Arms Support Command. December 18, 2007.

[23] Clausewitz, Carl Von. On War: Edited and Translated by Michael Howard and Peter Paret. New Jersey: Princeton University Press, 1976, rev. 1984.

[24] Corbet, Larry. "Sea Viking '06: The MAGTF And Distributed Operations" Presentation to students of Naval Post Graduate School, December 9, 2004.

[25] Cruse, Holk. Neural Networks as Cybernetic Systems. 2nd Ed. Accessed March 15, 2008. http://www.brains-minds-media.org/archive/290/index_html#d53e2282.

[26] De Mantaras, R. L. et al. "Retrieval, Reuse, Revision and Retention in Case-Based Reasoning." Knowledge Engineering Review, v 20, n 3, Sept. 2005, p 215-240.

[27] De Sanctis, G. and Gallupe, B. "Group Decision Support Systems: A New Frontier." Data Base, Winter 1985. p. 3-10.

[28] DeLaurentis, Daniel A. "Understanding Transportation as a System-of-Systems Design Problem," in proceedings of 43rd Aerospace Sciences Meeting and Exhibit, 10-13 Jan. 2005 AIAA-2005-0123.

[29] Department of Defense. Wikipedia: The Free Online Encyclopedia. Accessed August 4, 2008. http://en.wikipedia.org/wiki/United_States_Department_of_Defense

[30] Department of Defense: Office of Force Transformation. "Operational Sense and Respond Logistics." November 17, 2003.

[31] Department of the Navy. "A Concept for Distributed Operations." Headquarters United States Marine Corps, Washington D.C., April 25, 2005.

[32] Distributed Operations 2006 Capabilities and Enhancements Report as of 19 Jan 05.

[33] Distributed Operations. 2006 Naval research Advisory committee brief to the Assistant Secretary of the Navy, 23 June 2006.

[34] Enterprise Resource Planning from Wikipedia: The Free Online Encyclopedia. Accessed June 30, 2008. http://en.wikipedia.org/wiki/Enterprise_resource_planning.

[35] Espinosa, J., Vandewalle, J., Wertz, V. Fuzzy logic, Identification and Predictive Control. London: Springer-Verlang, 2005.

[36] Friedman, Thomas L. The World is Flat. New York: Farrar Straus and Giroux, 2005.

[37] Fuzzy Logic Toolbox. Mathworks Documentation Release 13, SP 2. Accessed September 12, 2008. http://www.mathworks.com/access/helpdesk_r13/help/toolbox/fuzzy/fuzzy.html .

[38] Garamone, J. "Casey: Army Must Adapt to Changing Threats." American Forces Press Service, November 17, 2007.

[39] Gilliver, Kate. "Review: Feeding an Army." JSTOR, Vol. 51, No. 2, 2001. p. 344-347.

[40] Gilman, B. MAJ. "Distributed Operations: Translating Tactical Capabilities into Operational Effects." Joint Military Operations Thesis: Naval War College, RI. February 13, 2006.

[41] Girratano, J. and Riley, G. Expert Systems. 3rd Ed. PWS Publishing Co., 1998.

[42] Global Security.org "Seabase." Accessed July 10, 2008. http://www.globalsecurity.org/military/systems/ship/seabase.htm.

[43] Global Security.org "Stryker Brigade Combat Team" Accessed January 8, 2008. http://www.globalsecurity.org/military/agency/army/brigade-ibct.htm.

[44] Google Images: Neuron. Accessed on 3 Jan 2007. http://www.web-books.com/elibrary/medicine/Physiology/Nervous/neuron.jpg.

[45] Gropman, Alan ed. The Big L: American Logistics in World War II. Washington DC: National Defense University Press, 1997.

[46] Haenni, Rolf. “Towards a Unifying Theory of Logical and Probabilistic Reasoning.” 4th International Symposium on Imprecise Probabilities and Their Applications, Pittsburgh, Pennsylvania, 2005.

[47] Harps, Leslie Hansen. “Military Logistics Shapes Up: As the Need for Efficiency in Military Operations Grows, Streamlining the Supply Chain becomes Imperative.” Defense Transportation Journal, Dec. 1, 2003.

[48] Herrera, L.J., Pomares, H., Rojas, I., Valenzuela, O., Gonz´alez, J., Awad, M. “MultiGrid-Based Fuzzy Systems for Time Series Forecasting: Overcoming the Curse of Dimensionality.” European Symposium on Artificial Neural Networks, Bruges (Belgium), 28-30 April 2004. p. 127-132.

[49] Hornik, K. “Approximation Capabilities of Multilayer Feed Forward Networks.” Neural Networks 4, 1991. p. 251–257.

[50] Hwang, C., Yoon, K. Multiple Attribute Decision Making: Methods and Applications: A State of the Art Survey. Berlin: Springer-Verlang, 1981.

[51] IC Tech Inc. “Fuzzy Control Methods and their Real System Applications.” Accessed December 28, 2007. http://www.ictech.com/Fuzzy%20Logic/tsld001.htm.

[52] Ilachinski, A. “EINSTein Summary Paper.” Kybernetes, Vol. 32, Iss. ½, 2003. pp. 38-77.

[53] Ilachinski, A. “Irreducible Semi-Autonomous Adaptive Combat (ISAAC): An Artificial-Life Approach to Land Warfare.” Center for Naval Analysis Research Memorandum 97-61.10, August 1997.

[54] Ilachinski, A. Artificial War. Singapore: World Scientific Publishing Co., 2004.

[55] Irie, B., Miyake, S. “Capabilities of Three-layered Perceptrons.” Proceedings of the IEEE International Conference on neural networks: prediction and system modeling. Technical Neural Networks, 1988. p. 641–648.

[56] Johnson, Carl and Schutte, Jeff. “Basic Regression Analysis for Integrated Neural Networks (BRAINN) Documentation V 2.1.” June 21, 2007.

[57] Joint Forces Quarterly. “An Interview with the Acting Director DOD, Office of Force Transformation.” Joint Forces Quarterly, Issue 42, 3rd Quarter 2006.

[58] Joint Logistics (Distribution) Joint Integrating Concept v. 1.0. The Joint Chiefs of Staff. February 7, 2006.

[59] Joint Publication 4.0: Doctrine for Logistic Support of Joint Operations. April 6, 2000.

[60] Josang, A. "Artificial Reasoning with Subjective Logic." Trondheim, Norway. Accessed June 20, 2008.
http://www.cse.unsw.edu.au/~ksg/acw97/Papers/Josang.ps.gz.

[61] Kartalopoulos, Stamatois V. Understanding Neural Networks and Fuzzy Logic. New York: IEEE Press, 1996.

[62] Keen, P. and Scott-Morton, M. Decision Support Systems: An Organizational Perspective, Reading. Massachusetts: Addison-Wesley, 1978.

[63] Kim, Kwang-Ho; Youn, Hyoung-Sun; Kang, Yong-Cheol. "Short-term Load Forecasting for Special Days in Anomalous Load Conditions using Neural Networks and Fuzzy Inference Method." IEEE Transactions on Power Systems Volume 15, Issue 2, May 2000. p. 559 – 565.

[64] Kirby, M. "A Methodology for Technology Identification, Evaluation, and Selection in Conceptual and Preliminary Aircraft Design." PhD Thesis, Georgia Institute of Technology, May 2001.

[65] Kirby, Michelle. "The 'How To' Guide for Response Surface Methodology." Version 1.0. ASDL, Georgia Institute of Technology. 13 March, 2003.

[66] Lawlor, M. "Data Farming Cultivates New Insights." Signal Magazine, June 2005. Accessed November 22, 2008.
http://www.afcea.org/signal/articles/anmviewer.asp?a=975&print=yes.

[67] Leake, D. Case-Based Reasoning: Experiences, Lessons & Future Directions. Massachusetts: MIT Press, 1996.

[68] Lewe, J., "An Integrated Decision-Making Framework for Transportation Architectures: Application to Aviation Systems Design." PhD Thesis, Georgia Institute of Technology, 2005.

[69] Lisi, F., and Schiavo, R. A. "A Comparison between Neural Networks and Chaotic Models for Exchange Rate Prediction." Computational Statistics and Data Analysis, Vol. 30, 1999. p. 87-102.

[70] Little, J. D. C. "Models and Managers: The Concept of a Decision Calculus, Management Science." Volume 16 Issue 8, 1970. p. 466 – 485.

[71] Logistics. Wikipedia: The Free Online Encyclopedia. Accessed January 20, 2008.
http://en.wikipedia.org/wiki/Logistics.

[72] Mahdavi, N., Menhaj, M. B., and Barghinia, S. “Short-Term Load Forecasting for Special Days Using Bayesian Neural Networks.” Power Systems Conference and Exposition, 2006. PSCE '06. 2006 IEEE PES. Oct. 29 2006-Nov. 1 2006. p. 1518 – 1522.

[73] Maier, Mark W., "Architecting Principles for System of Systems," Systems Engineering, Vol. 1, No. 4, 1998. p. 267-284.

[74] Makridakis, S., Wheelwright, S., McGee, V. Forecasting: Methods and Applications. 2nd Ed. New York: Wiley and Sons, 1983.

[75] Marine Expeditionary Unit. Wikipedia: The Free Online Encyclopedia. Accessed November 4, 2008. http://en.wikipedia.org/wiki/Marine_Expeditionary_Unit.

[76] Martelli, A. Strategic Analysis Through Scenarios: From the Macro Microenvironment: Theories and Methods. Milano (Italy): ETAS Libri, 1992.

[77] Mathworks MatLab Electronic Users Manual. Topic: *lhsdesign.* Last Accessed August 7, 2008.

[78] Mathworks, “Foundations of Fuzzy Logic”. Accessed October 19, 2007. http://www.mathworks.com/access/helpdesk_r13/help/toolbox/fuzzy/fuzzy.html.

[79] Matlab Tutorial for closing a GUI. Accessed October 10, 2008. http://www.blinkdagger.com/matlab/matlab-gui-tutorial-close-gui-confirmation.

[80] Mavris, D. N., DeLaurentis, D. A., Bandte, O., Hale, M. A. “A Stochastic Approach to Multi-disciplinary Aircraft Analysis and Design,” in AIAA-98-0912, Presented at the 36th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 12-15, 1998.

[81] McGhee, J., Grimble, Michael J., Mowforth, Peter. Knowledge Based Systems for Industrial Control. United Kingdom: Michael Faraday House, 1990.

[82] McIntosh, G, Galligan, D., Anderson, M. Lauren, M. Map Aware Non-Uniform Automata (MANA) Version 4 Users Manual. New Zealand Defence Technology Note, 2007.

[83] McIntosh, G., Galligan, D., Anderson, M., Lauren, M. “Map Aware Non-Uniform Automata (MANA) Users Manual. DTA Technical Note 2007/3, NR1465, May 2007.

[84] McKay, M.D. “Latin Hypercube Sampling as a tool in Uncertainty Analysis of Computer Models.” Proceedings of the 1992 Winter Simulation Conference, 1992.

[85] McVey, W.J. “Control System Requirements for Future Propulsion Systems.” Pratt & Whitney, Presentation at Georgia Institute of Technology, December 2000.

[86] Mears, Gary H and Kim, Ted. “Logistics: The Way Ahead.” Joint Forces Quarterly, Issue 4, 1994.

[87] Michalwicz, Z. Genetic Algorithms + Data Structures = Evolution Programs. Heidelberg: Springer-Verlag, 1996.

[88] Mietzner, D., Reger, G. “Paper 3: Scenario Approaches – History, Differences, Advantages, and Disadvantages.” Presented at the EU-US Seminar: New Technology Foresight, Forecasting, and Assessment Methods. Seville, May 13-14, 2004.

[89] Monte Carlo Simulation. Accessed January 17, 2008. http://en.wikipedia.org/wiki/Monte_Carlo_method, July 2008.

[90] Murdock, J.W., Aha, D.W., & Breslow, L.A. “Case-based argumentation via process models”. Proceedings of the Fifteenth International Conference of the Florida. Artificial Intelligence Research Society. St. Augustine, FL: AAAI Press, 2003.

[91] Murdock, J.W., Aha, D.W., and Breslow, L.A. Lecture Notes in Computer Science, Vol 2689/-1. Title: “Assessing Elaborated Hypotheses: An Interpretive Case Based Reasoning Approach.” p. 332–346. Berlin Heidelberg : Springer, 2003.

[92] MV-22 Osprey. Wikipedia: The Free Online Encyclopedia. Accessed November 4, 2008. http://en.wikipedia.org/wiki/MV-22_Osprey.

[93] Napoleon's Invasion of Russia. Accessed Dec 17, 2007. http://www.geographia.com/russia/rushis05.htm.

[94] Neapolitan, Richard E. Probabilistic Reasoning in Expert Systems. New York: Wiley & Sons, 1990.

[95] Neural Network Toolbox. Mathworks Documentation Release 13, SP 2. Accessed October 28, 2007. http://www.mathworks.com/access/helpdesk_r13/help/toolbox/nnet/nnet.html.

[96] NIST/SEMATECH. “e-Handbook of Statistical Methods.” Accessed August 20 2008. http://www.itl.nist.gov/div898/handbook/.

[97] Office of Force Transformation. Wikipedia: The Free Encyclopedia. Accessed October 7, 2007. http://en.wikipedia.org/wiki/Office_of_Force_Transformation.

[98] Office of Management and Budget. "Table 3.1: Outlays by Superfunction and Function: 1940-2012," Historical Tables, Budget of the United States Government, FY2008. Accessed April 25, 2008. http://origin.www.gpoaccess.gov/usbudget/fy08/hist.html.

[99] Oklahoma State University: Office of Intellectual Property. Accessed January 15, 2008. http://www.vpr.okstate.edu/ipm/basics/threetest.html.

[100] Operational Sense and Respond Logistics: Co-Evolution of an Adaptive Enterprise Capability. Office of Force Transformation. 17 Nov 2003.

[101] Pace, D. "Modeling and Simulation Verification and Validation Challenges." John Hopkins APL Technical Digest, Vol. 25, No. 2 (2004). p. 163 - 172.

[102] Pagonis, William G. and Cruikshank, Jeffrey L. Moving Mountains: Lessons in Leadership and Logistics from the Gulf War. Massachusetts: Harvard Business School Press, 1992. p. 5.

[103] Paparone, C. "Operations on the Cusp of Postpositivism." Small Wars Journal, May 28, 2008. Accessed September 3, 2008. http://www.smallwarsjournal.com.

[104] Pearl, J. Probabilistic Reasoning in Intelligent Systems. California: Morgan-Kauffman, 1988.

[105] Pena Reyes, Carlos. Coevolutionary Fuzzy Modeling. Germany: Springer, 2004.

[106] Potter, M.A. "The Design and Analysis of a Computational Model of Cooperative Coevolution." PhD Dissertation: George Mason University, 1997.

[107] Potter, M.A. and DeJong, K. A. "Cooperative Co-evolution: An Architecture for Evolving Co-adapted Subcomponents." Evolutionary Computation, Vol. 8, No. 1, 2000. p. 1-29.

[108] Priddy, Kevin L and Keller, Paul E. Artificial Neural Networks: An Introduction. Washington: SPIE Press, 2005.

[109] Project Albert Website. Accessed November 17, 2007. http://projectalbert.org/.

[110] Pudas, T. "The Next Step in Transformation." Defense Systems. February 2007.

[111] Reed, Russell D, and Marks, Robert J. Neural Smithing: Supervised Learning in a Feed-forward Artificial Neural Network. Massachusetts: MIT Press, 1999.

[112] Richter, M. M. and Aamodt, A. "Case-Based Reasoning Foundations. Knowledge Engineering Review, v 20, n 3, Sept. 2005, p 203-207.

[113] Saxena, A. “Knowledge Based Architecture for Integrated Conditioned Based Maintenance of Engineering Systems.” PhD Thesis, Georgia Institute of Technology, May 2007.

[114] Sea Viking 2006 Distributed Operations Wargame Information CD. November 2004.

[115] Seabasing Joint Integrating Concept (JIC), Version 1.0. August 1, 2005, 5.

[116] Seabasing Logistics Enabling Concept. December 2006.

[117] Shafer, G. and Pearl, J. Readings in Uncertain Reasoning. California: Morgan-Kauffman, 1990.

[118] Shonkwiler, R., “Math 4225 Class Notes.” School of Mathematics, Georgia Institute of Technology, 2004.

[119] Shrader, Charles R. U.S. Military Logistics, 1607-1991, A Research Guide. New York: Greenwood Press, 1992. p. 9.

[120] Sindiy, Oleg V.; DeLaurentis, Daniel A.; Akaydin, Kirk A.; Smith, Donald A.; Improved Decision Support in Space Exploration via System of Systems Analysis.” IEEE International Conference for SOSE on 16-18 April 2007. p. 1-6.

[121] Soermo, F., Cassens, J., and Aamodt, A. ”Explanation in Case-Based Reasoning - Perspectives and Goals.” Artificial Intelligence Review, v 24, n 2, Oct. 2005, p 109-143.

[122] Soft Computing Homepage. Accessed December 10, 2007. http://www.soft-computing.de/pr.html.

[123] Sprague, “A Framework for the Development of Decision Support Systems.” MIS Quarterly, 4. p. 1-26.

[124] System of Systems. Wikipedia: The Free Online Encyclopedia. Accessed December 10, 2007. http://en.wikipedia.org/wiki/System_of_systems.

[125] Systems Engineering Process. Wikipedia: The Free Online Encyclopedia. Accessed December 10, 2007. http://en.wikipedia.org/wiki/Image:Systems_Engineering_Process.jpg.

[126] Systems Engineering. Wikipedia: The Free Online Encyclopedia. Accessed December 10, 2007. http://en.wikipedia.org/wiki/Image:SE_Activities.jpg.

[127] Thacker Jr., D. “The Joint Interdependence of Distributed Operations.” U.S. Army War College: Carlisle Barracks, PA. January, 21, 2005.

[128] The National Defense Strategy of the United States of America, March 2005. Accessed April 17, 2008. http://www.globalsecurity.org/military/library/policy/dod/nds-usa_mar2005.htm

[129] The National Military Strategy of the United States of America, March 2005. Accessed April 17, 2008. http://www.globalsecurity.org/military/library/policy/dod/nds-usa_mar2005.htm.

[130] The National Security Strategy of the United States of America, September 2002. Accessed April 17, 2008. http://www.whitehouse.gov/nsc/nss.html.

[131] Thorpe, George C. Pure Logistics. Washington DC: National Defense University Press, 1987.

[132] Tiron, Roxana. "Seabase Hurdles: Deployment of Seabases faces technical, budgetary challenges." National Defense Online, Jan 1, 2005.

[133] Top Down vs. Bottom Up Planning. Accessed January 6, 2009. http://www.brighthub.com/office/project-management/articles/8542.aspx.

[134] Torchbearer National Security Report. "The New Paradigm: Bringing US Army Logistics into the 21st Century." Published by AUSA, April 2004.

[135] TRANCCOM: Deployable Distribution Command and Control Brief. September 2007.

[136] U.S. Army. "Army Programs Decision Risk Analysis", CASCOM, Fort Lee, VA, 1990.

[137] Upton, S. C. "Users Guide: Old McData-The Data Farmer ver. 1.0.", March 5, 2008.

[138] Vogl, T.P.; Mangis, J.K.; Rigler, A. K.; Zink, W.T.; and Alkon, D.L. "Accelerating The Convergence of The Back-propagation Method." Biological Cybernetics, Vol. 59, 1988. p. 257-263.

[139] Vose, M. D. The Simple Genetic Algorithm. Massachusetts: MIT Press, 1999.

[140] Winters, M. F. and Linkage Inc. "Top-down vs. Bottom-up: Successfully Implementing Diversity Initiatives." Accesses December 29, 2008. http://www.workinfo.com/free/Downloads/99.htm.

[141] Yen, J. and Langari, R. Fuzzy Logic: Intelligence, Control, and Information. New Jersey: Prentice Hall, 1999.

[142] Zadeh, L. A. "Fuzzy Logic and the Calculi of Fuzzy Rules, Fuzzy Graphs, and Fuzzy Probabilities." Computers and Mathematics with Applications, 37, 1999. p. 35.

[143] Zadeh, L. A. “Fuzzy Logic, Neural Networks, and Soft Computing.” Communications of the ACM, Vol. 37, No. 3. March 1994. p. 77-84.

[144] Zadeh, L. A. “Fuzzy Sets.” Information and Control, 9, 1965.

[145] Zadeh, L.A. "The Concept of Linguistic Variable and its Application to Approximate Reasoning - Part I." Information Sciences, 8, 1975. p. 199-249.

[146] Zadeh, L.A. "The Concept of Linguistic Variable and its Application to Approximate Reasoning - Part II." Information Sciences, 8, 1975. p. 301-357.

[147] Zadeh, L.A. "The Concept of Linguistic Variable and its Application to Approximate Reasoning - Part III." Information Sciences, 9, 1975. p. 43-80.

[148] Zhang ,Guoqiang; Patuwo, Eddy B.; Hu, Michael Y. “Forecasting with Artificial Neural networks: The State of the Art.” International Journal of Forecasting, 14, 1998. p. 35–62.

VITA

Jesse Stuart Hester was born in Hawaii in September of 1972. During his senior year in High School, Mr. Hester enlisted in the United States Army Reserve. Upon completion of High School, Mr. Hester attended Basic Training and Advanced Individual Training. Shortly thereafter, Mr. Hester was accepted into the United Sates Military Preparatory School (USMAPS) located in Ft. Monmouth, New Jersey. After a year of education and training at USMAPS he was accepted into the Class of 1996 at the United States Military Academy or West Point. At West Point, then Cadet Hester, pursued a Bachelors of Science Degree in Mechanical Engineering and became Airborne and Air Assault qualified. Graduating on June 1, 1996, 2LT Hester entered the Army and was branched into the Engineer Corps. Successful completion of the Office Basic Course at Ft Leonard Wood in Missouri was followed by attendance to Ranger School and Pathfinder School. He resigned his commission from the Army Reserve as a Captain in 2007. His assignments consisted of:

- Vicenza, Italy: Heavy and Combat Engineer Platoon Leader with the Southern European Task Force (SETAF) Lion Brigade.
- Ft Leonard Wood, Missouri: Basic Rifle Marksmanship Committee Chief and S-3 for the Training Support Battalion, 3rd Training Brigade.
- Ft. Gillem, Georgia: Engineer Team Officer in Charge for the 1/347th (CS/CSS).

After leaving active duty, Mr. Hester worked for Orbital Sciences Corporation in Dulles Virginia for three years. In October of 2001 Mr. Hester received his private pilot’s license. In pursuit of achieving his goal of becoming an astronaut for NASA, Mr. Hester returned back to academia in 2003 to attend The Georgia Institute of Technology in Atlanta, Georgia with the goal of earning a PhD in Aerospace Engineering. Mr. Hester also pursued and was conferred a Masters of Science in Aerospace Engineering and a Masters in Business Administration, both in May of 2006.