AGENT BASED SIMULATION

SEAS EVALUATION OF DODAF ARCHITECTURE

THESIS

Gregory V. DeStefano, Captain, USAF

AFIT/GOR/ENS/04-05

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/ENS/04-05

AGENT BASED SIMULATION SEAS EVALUATION OF DODAF ARCHITECTURE

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Gregory V. DeStefano, B.A.

Captain, USAF

March 2004

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT/GOR/ENS/04-05

AGENT BASED SIMULATION SEAS EVALUATION OF DODAF ARCHITECTURE

Gregory V. DeStefano, B.A.

Captain, USAF

Approved:

//Signed//	9 March 04
Dr. John O. Miller Thesis Advisor	Date
//Signed//	10 March 04
Dr. David R. Jacques Committee Member	Date

AFIT/GOR/ENS/04-05

Abstract

With Department of Defense (DoD) weapon systems being deeply rooted in the command, control, communications, computers, intelligence, surveillance, and reconnaissance (C4ISR) structure, it is necessary for combat models to capture C4ISR effects in order to properly assess military worth. Unlike many DoD legacy combat models, the agent based model System Effectiveness and Analysis Simulation (SEAS) is identified as having C4ISR analysis capabilities. In lieu of requirements for all new DoD C4ISR weapon systems to be placed within a DoD Architectural Framework (DoDAF), investigation of means to export data from the Framework to the combat model SEAS began. Through operational, system, and technical views, the DoDAF provides a consistent format for new weapon systems to be compared and evaluated. Little research has been conducted to show how to create an executable model of an actual DoD weapon system described by the DoDAF. In collaboration with Systems Engineering masters student Captain Andrew Zinn, this research identified the Aerospace Operation Center (AOC) weapon system architecture, provided by the MITRE Corp., as suitable for translation into SEAS. The collaborative efforts lead to the identification and translation of architectural data products to represent the Time Critical Targeting (TCT) activities of the AOC. A comparison of the AOC weapon system employing these TCT activities with an AOC without TCT capabilities is accomplished within a Kosovo-like engagement (provided by Space and Missile Center Transformations Directorate). Results show statistically significant differences in measures of effectiveness (MOEs) chosen to compare the systems. The comparison also identified the importance of data products not available in this incomplete architecture and makes recommendations for SEAS to be more receptive to DoDAF data products.

Acknowledgements

I would like to thank Dr. J.O. Miller for his direction and guidance in this thesis effort. His knowledge of combat simulation and analysis techniques were instrumental in my ability to quantify my findings in a non-traditional manner. Also, I am grateful to him for enduring the countless revisions to my document. I would also like to thank Mr. Robert Weber in his seemingly endless knowledge and resources in agent based simulation and complex systems.

This effort would have been bust if not for my research partner Drew Zinn. Drew, the countless hours working on a best method to transfer data from your "system engineering world" to my "combat modeling world" paid off.

Finally, I would like to thank all my friends, classmates, and family who have been there to understand all the late nights at the lab, listening to all the thesis speak, and for always having a sense of humor when it was most needed.

Gregory V. DeStefano

Table of Contents

List of Figures

Figure | Page

List of Tables

AGENT BASED SIMULATION
SEAS EVALUATION OF DODAF ARCHITECTURE

1. Introduction

1.1 Background

Air power theory suggests that the effects of quick strikes and global reach propagate throughout an opponent's military. The propagation is expected to yield catastrophic output or strategic effects [3]. All U.S. military forces expect to take advantage of similar effects by attacking tactical, operational, and strategic targets in concert. This theory rests largely upon an observe, orient, decide, and act (OODA) model of warfare. Unfortunately, many current war gaming, training, and analysis simulations' models of war are built upon "Cold War" doctrine which do not support current network centric and asymmetric warfare. The OODA model of warfare places significant stock in the contribution of Command, Control, Communication, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR).

Operational studies have shown agent based simulation utilizing this OODA model of war as a competent method to gain insight to the military value of C4ISR systems. The gain being the ability to match and evaluate the performance of a system with its concept of operation in a multitude of scenarios. Slowly all services are beginning to integrate these models in to their respective repertoire of simulation models. The Air Force standard analysis toolkit (AFSAT) is an Air Force approved collection of models and simulations (M&S) tools which are to support acquisitions and operational decisionmaking. Included in the AFSAT, the only agent based simulation, is the combat model System Effectiveness Analysis Simulation (SEAS). SEAS was designed by the Air Force Space and Missile Center (SMC) to specifically address the aforementioned OODA loop model of war and the functional contribution of C4ISR systems [31].

In an effort to improve the process of acquiring and implementing these C4ISR systems, the DoD requires the use of systems architectures. This was in response to lessons learned during the Gulf War which eventually led to a congressional mandate in the form of the Clinger-Cohen Act [32]. To improve and standardize these architectures, the DoD has created an Architectural Framework (know as the DoDAF for DoD Architectural Framework). The DoDAF divides the description into three views: operational, system, and technical. These views provide a broad overview of the system, and also views yielding specific interconnections supporting warfighting functions. Each view consists of several architectural "products" that are named according to their view (i.e. OV-1, SV-4, etc.). The Air Force Chief Architect's office has advocated several possible uses for these architectures, one of which is, "Military Worth Analysis" using M&S (AF-CIO/A at https://cao.hanscom.af.mil/af-cio.htm).

1.2 Research Problem

The current guidance on the use of system architectures is clear, but unproven [32]. Translating data from system architecture views in order to evaluate the dynamic effects of a C4ISR system would be of great assistance in an Analysis of Alternatives (AoA). As mentioned SEAS has been built for and is capable of capturing the effects of the proposed system in a combat scenario. Once, a specific instance of a C4ISR system represented in DODAF is translated then general algorithms may be able to be created affording evaluation of any architecture while still early in the acquisition cycle.

1.3 Research Objective

The objective of this research is to investigate the means of exporting data from operation, system, and technical architecture products of the DODAF to the agent-based combat model SEAS. This is to be accomplished such that the combat model captures proper measures of effectiveness to evaluate the represented systems' worth. Also, to provide consistent translation of performance and concept of operations (ConOps) parameter definitions. A secondary objective is to comment on the maturity of the architecture evaluation process using dynamic simulation, and what needs to be done to improve the process.

1.4 Thesis Overview

Chapter 2 is divided into 4 main sections. The first section provides background on traditional combat M&S used in the DoD. The second section describes agent based simulation and the model of war they are built upon. Also, an overview of the combat model SEAS is given. A third section provides background on the structure and intent of C4ISR architectures. Finally, a fourth section reviews current research in the dynamic modeling of architectures. Chapter 3 first provides an overview of the architectural products used in this study. Then we give a description of the scenario the system is evaluated in. Next, translations of communications, activities, and general attributes from the architectural products to SEAS is given. Finally, verification, validation, and analysis issues are discussed. Chapter 4 provides the development of the analysis and numerical results of measures of effectiveness. Lastly, a summary of the military utility analysis substantiating the use of architectures represented in DoDAF as plausible source documentation for M & S is presented. Also, limitations on the study due to the maturity of the architecture and the combat model are given in Chapter 5.

2. Literature Review

2.1 Introduction

This research is a collaborative effort investigating the means of exporting data from architectural products based on the Department of Defense Architectural Framework (DoDAF) to an agent-based combat model (ABCM). The ABCM chosen for this role is the System Effectiveness Analysis Simulation (SEAS). This transition of data allows Military Utility Analysis (MUA) of the weapon system depicted in the architecture.

Current Department of Defense (DoD) Modeling and Simulation practices will be reviewed with focus on how SEAS provides effective MUA analysis for C4ISR effects of a weapon system. Also, this literature review develops the concept of the DoDAF and its implementation in the recently revised acquisition process. Finally, relevant research involving dynamic modeling of architectures is presented.

2.2 DoD Combat Modeling & Simulation

The (DoD) uses a vast number of models with the majority focused on some aspect of combat. According to the Defense Modeling and Simulation Office (DMSO) current uses of combat models include training, analysis and acquisition. In this review focus is placed on exploring the proper "level" and type of combat model to be used in analysis and acquisition. First, a discussion of major underlying assumptions used in many combat models will be given. Next, the "levels" of combat models used by the DoD will be discussed. Finally, in efforts to lay foundation for ABCM treating war as a Complex Adaptive System (CAS) is visited.

2.2.1 Lanchester

Over the course of history combat models have been built upon the given rules of warfare of the time. Widely used equations to drive modern warfare models can be associated with Frederick William Lanchester. Lanchester, a British engineer and inventor assumed two scenarios, single combat (hand-to-hand) and theater campaigns (armies) [14]. The first scenario involves a simple linear relationship between the number of troops and the loss rate. The second scenario involves a proportional relationship between loss rate and number of enemy firers. Thus, these two models have been coined Lanchester's linear and square laws.

The basic approach is to write a differential equation that equates the loss rate of two homogeneous forces via two factors; (1) the size or number of opposing troops and (2) the effectiveness of the killing power of each troop.

The generic Lanchester Equations of modern warfare "aimed fire" are stated below.

$$\frac{dX}{dt} = -aY$$
$$\frac{dY}{dt} = -bX$$

Where:

X = number of blue forces
Y = number of red forces
a= effective firing rate of red
b = effective firing rate of blue

Demonstrating the principle of concentration of force was the original intent of the above coefficients (a,b). These Lanchester equations can answer questions in homogeneous combat like determining the number of survivors, how long will the battle last, and who won the fight. Rarely will combat ever consist of homogenous (individual elements of the forces have identical characteristics)forces. A simple example is the introduction of tanks into a ground battle. The tank will have a greater killing power then the troops it is fighting with. Refinements to address this attrition problem and others have been developed.

Not all deficiencies with combat models lie within the attrition processes. Other issues include range dependency, replacement of forces, C4ISR, and little or no portrayal of logistical aspects. These issues coupled with attrition are necessary to more accurately portray combat. Significant enrichments and enhancements have been made to accommodate these and other unmentioned shortcomings. For over eighty years Lanchester equations have remained a well used tool in the evaluation of combat. However, as doctrine, tactics, and politics change constant revision will be needed to this equation based model of combat.

The Lanchester equations can be considered the model of war, and the evaluation of this model over various time steps yields answers we are searching for. In fact, the DoD defines simulation as a method for implementing a model over time or as a technique for testing, analysis, or training in which real-world systems are used, or where real-world and conceptual systems are reproduced by a model [9]. Lanchester equation models provide some structure, components, and interactions of the war, and simulation evaluates the war over time.

Computer simulations may use Lanchester expressions "locally" (i.e., for attrition estimates within a given time interval), but the coefficients of those equations change from time step to time step as conditions of terrain, defender preparations, and many other factors change. Furthermore, good computer simulations recognize that the losing side may choose to break off battle rather than be annihilated [6]. Unfortunately, even when considering the enrichments of Lanchester equations they will not provide feedback to issues like when the losing side will decide to "break" off the battle. To evaluate this break some human intevention of the simulation is necessary.

However, Lanchester equations integrate well with the "piston driven" warfare we executed in World Wars I and II and planned on fighting against Warsaw Pact forces. For many years we expected our battles to be fought out exclusively on lines (Forward Edge of the Battle Area - FEBA) with Red forces massed on one side and Blue on the other. Troops and equipment are modeled to engage along FEBA, and never drift too far horizontally. The parallel strips of forces, pistons, are assessed for size, and firepower so that attrition can be calculated. This structured war allowed Lanchester equations to predict attrition quite nicely.

2.2.2 Aggregation

Within each piston there may be many individual troops and a large amount of equipment. When modeled in this manner, troops and equipment are called entities which may have associated values (attributes). When large scale scenarios are to be evaluated it is common to lump similar entities together. This lumping is referred to as aggregation. Aggregation is defined by DoD 5000.59-M as "the ability to group entities while preserving the effects of entity behavior and interaction while grouped" [10]. Aggregation is not limited to lumping soldiers into a unit, but also resources may be aggregated to the unit, or even the brigade level. This may be done for desired output measures, or the shear number of entities may not be able to be computed in a reasonable amount of time.

Capturing output for all individual force interactions on the battlefield may not be possible or even the intent of a given model. DoDs combat model hierarchy(Figure 2.1) categorizes combat models based on resolution and aggregation. Resolution is defined as: "the degree of detail and precision used in the representation of real world aspects in a model or simulation" [9]. A high resolution combat model (engineering level) includes the physics of an engagement between a small number of platforms. These models typically yield some Measure of Performance(MOP) of a system versus a threat (e.g. the ability of a stealth fighter to pass over enemy radar without

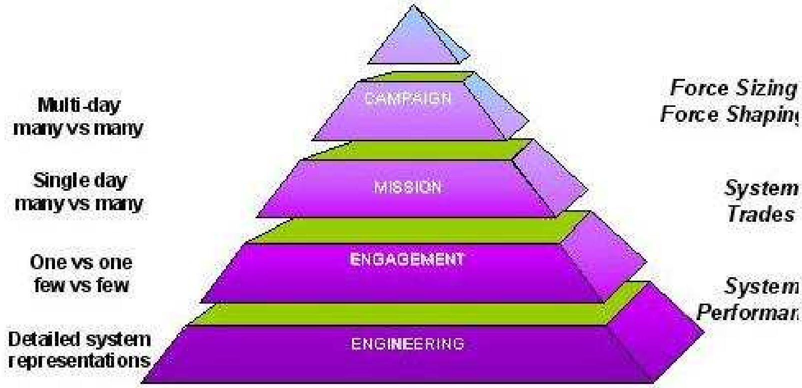

Figure 2.1 Combat Model Hierarchy

detection). Aggregate, low-resolution combat models(campaign level) are needed in cases where the historical data available is of aggregated nature. Also, command and control (C2) decisions often require information at an aggregate level [15].

Aggregation is intended to preserve the effects of entity behavior and interaction while grouped, but little theory or science is present in most approaches that have been taken to aggregation in combat modeling [15]. Conclusions of a companion paper of Hillestad on theoretical issues of aggregation state that "aggregation and disaggregation cannot be done arbitrarily and that fairly strong requirements are necessary to obtain consistent high- and low-resolution models" [16]. Disaggregation means the evaluation of the number of each entity type attrited when the aggregate force has been reduced. Typically, it is of interest to preserve effects or be consistent from the high- to low-resolution models. According to Caldwell, "The high-resolution model has some face validity because its detailed process structures follow real actions and events closely. Thus, being consistent with a high-resolution model would be an advantage for an aggregated simulation" [4].

2.2.3 Problems with Aggregation

Despite calibration and other techniques to maintain consistency, aggregated models used by the DoD lose some information. Hartman addresses the following consequences of aggregation. Aggregating individual combatants into groups result in lost information of the individuals actions. What each agent is doing at any given time is not known. Also, loss of information about event sequences occurs from not keeping track of individual entity actions [14]. Of course, action is taken to compensate for this information loss. Typically, this is done by modeling combat processes using average behavior rather than individual behavior. Overall, it is critical to use aggregation techniques that do not directly affect the output measures that are to be characterized by the simulation. This means if averaging behavior directly affects an output measure of the simulation some other method needs to be employed.

2.2.4 C4ISR Effects

Information Superiority (IS) has been identified as a key element of U.S. doctrine. Emphasis is placed on generating and protecting the flow of information. Advanced C4ISR systems are intended to provide IS and deny the enemy this capability. IS is seen as an enabler for improved, rapid, and smart command decisionmaking or so-called Decision Superiority [12].

The flow of information for decision making is often looked at as a BOYD (Col. John Boyd, USAF (Ret)) OODA loop. The OODA (Observe Orient Decide Act) loop describes the thought process of a single warrior, a pilot, or a commander. Each member observes his surroundings through collected information via sight, sensors, advisors, etc. He then orientates himself based on this information. Next, some decision of action to be taken is reached. Finally, action is taken to attack, maneuver, or evade the enemy. It is traditional wisdom that the advantage goes to he who processes his OODA loop faster than the enemy. This can be confirmed by computer simulation studies or real life action that Col Boyd observed, such as when U.S. pilots out maneuvered faster Russian Migs, in turn having the advantage of taking the next action before the enemy. He created the OODA loop from this experience, and it has been applied across military and business processes.

The contributions of C4ISR systems should be evaluated not just at the *technical level* (e.g., how quickly and accurately systems collect and transmit information), but also at a higher level that can be related to emerging joint doctrine and operational-level command decision-making [12]. In other words it is important to determine how the information supplied by C4ISR systems helps to achieve operational advantages that in turn result from dynamic command decision making. This typically can not be done via traditional DoD combat simulations. Ilachinski [18] states that Lanchester attrition calculations do not account for spatial variation of forces or advantage of maneuver. C4ISR effects are also casualties of aggregation as seen in studies by RAND, Space and Missile Center (SMC), and other DoD agencies. Also, the need for new measures and metrics that incorporate the effectiveness of C4ISR systems, procedures, and equipment and their effect on combat outcome is emphasized in work by Perry [30]. Integration of DoD M&S is called upon as one of the key enablers to produce these metrics. Not to discount the need and utility of equation based models but, a different model of war is needed to capture and illuminate the effects of C4ISR described above.

2.3 Modeling War as a complex adaptive system

Enemies adapt and learn our behaviors, and in turn we do the same. As per Air Force Basic Doctrine, war is considered “complex and chaotic” and “war is not waged against an inanimate or static object, but against a living, calculating enemy” [1]. This being conceded in our doctrine, capturing these behaviors of war in combat models seems imperative.

The doctrine conflicts with the rigid mathematical rules set in many of the DoD combat models (e.g. Lanchester-Based models). Although stochastic and deterministic mathematical based models do have solid purposes they are too rigid to capture "flexible and situational principles of war that are to be applied to tactical decisions" [29]. Modeling war as a Complex Adaptive System offers insight to elements of war not captured by the traditional style of model, such as C4ISR effects.

2.3.1 Complex Adaptive Systems

A Complex Adaptive System (CAS) is defined as a set of elements that are interconnected so that changes in some elements, or their interrelations, produce changes in other parts of the system. Also, the entire system demonstrates properties and behaviors that are different from those of the individual parts [21]. An example is attacking a Center of Gravity (COG) or a commanders OODA loop. This can be considered a change in some *element* of the system in hope to change the behavior of the whole system. If the entities in war can be characterized as a set of elements that are interconnected such that the aforementioned results can be observed, we are on the way to modeling war as a CAS.

Modeling a CAS follows a solid mathematical backing of complexity, and chaos. A good development of this can be found in Ilachinski [19]. There are also other defining points of a CAS including decentralized control, self-organization, and non-linear interaction. Since warfare commonly demonstrates these qualities it has been asserted that combat should be able to follow the same methodological course of study as any other complex adaptive system [18].

2.3.2 Agent Based Modeling

Agent based modeling claims to allow adaptive decision making by entities. This is accomplished by allowing individual entities to act autonomously, governed by its own set of rules. ABM has been proposed for many situations involving a large number of heterogeneous individuals, such as vehicles and pedestrians in traffic, people in crowds, artificial characters in computer games, agents in financial markets, and humans and machines on battlefields [28].

Agent-based simulations are well suited for testing hypotheses about the origin of observed emergent properties in a system. This can be done simply by experimenting with sets of initial conditions at the micro-level necessary to yield a set of desired behaviors at the macro-level. On the other hand, they also provide a powerful framework within which to integrate ostensibly "disjointed" theories from various related disciplines. For example, while basic agent-agent interactions may be described by simple physics and sociology, the internal decision-making capability of a single agent may be derived, in part, from an understanding of cognitive psychology. This understanding forms the building block of most models of complex adaptive nature [5].

Combat agent-based models offer an opportunity to analyze, the behavior and interactions between the participating entities of war, where traditional models typically only analyze the performance of specific weapons or sensors. In other words, we shift our attention from analyzing the performance of pieces of equipment to how different modes of operation may alter the outcome of a particular combat or peacekeeping operation, or how the C2 system utilizes information and acts upon it [23].

2.3.3 Agent Based Models in the DoD

The DoD has recently incorporated ABM into its combat modeling library. Evidence of ABM can be seen in studies conducted by all four major services.

The NAVY utilizes Agent Based simulation with the Naval Simulation System (NSS). NSS is the primary model for supporting network centric Fleet Battle Exercises (http://www.metsci.com/ssd/nss.html). Network centric warfare enables a shift from attrition-style warfare to a much faster and more effective warfighting style characterized by the new concepts of speed of command and self-synchronization [2]. NSS is capable of capturing the self-synchronization of warriors and the effects of speed of command in combat due to its agent qualities and CAS model of war.

The Marine Corps relies on JIVES (Joint Integrated Visualization Environment Simulation) for insight to such issues as urban conflict. Also, ISAAC (Irreducible Semi-Autonomous Adaptive Combat) has been created in part by the Marine Corps to evaluate how certain aspects of land combat can be viewed as emergent phenomena resulting from the collective, nonlinear, decentralized interactions among notional combatants [5].

As part of the Air Force Standard Analysis Tool Kit(AFSAT) SEAS is used for a wide range of analysis. The Army has also taken advantage of SEAS in a study to asses the value of Information Superiority for ground forces [12].

2.3.4 SEAS

SEAS is a stochastic, mission-level model designed for use in evaluation of the military utility of airborne and space-based communications and intelligence, surveillance, and reconnaissance (ISR) assets [27].

As mentioned previously, aggregation of highly detailed system level simulations provide data for mission level models like SEAS. However, SEAS utilizes a vertical aggregation method (reduction of the overall number of entities in the system) as to previously discussed horizontal aggregation where like entities are lumped together. This method allows the preservation of the essential configuration of forces in space and time that is critical in assessing the impact of ISR and information dominance on combat outcomes [17].

Each agent in SEAS runs a parallel execution thread with interactions adjudicated on a time tic by time tic basis. Time increments are one minute. In between time steps a set of processes are carried out. Figure 2.2 shows the event processing

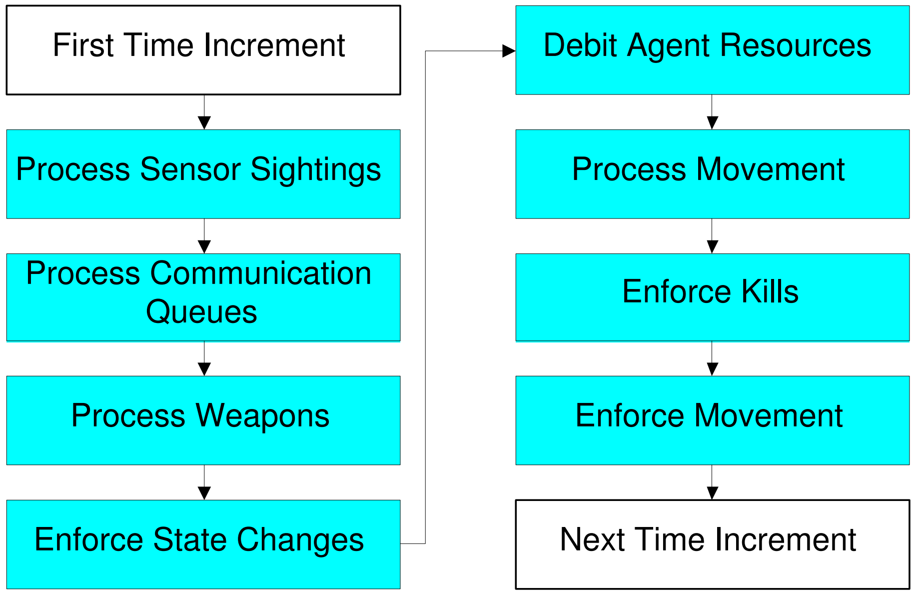

Figure 2.2 SEAS Event Processing

logic with various types of information processed, changes made to the state of the system, and then enforcement of individual agent actions. This process in SEAS is based on the Boyd OODA loop.

Figure **??** highlights the hierarchical nature of a forces' assets, and a high level description of interactions between agents. Two types of agents in SEAS (unit and platform) interact within an environment via its owned equipment. An example of a unit in SEAS is an Aerospace Operations Center (AOC). This unit agent may own other unit agents such as a Fighter Squadron. Within the unit, agent platform agents such as a F-16 fighter reside. More detail of the interaction and execution of agent actions is available at www.teamseas.com. Each platform agent may have a simple rule set which take precedences over commands passed from a unit agent. These rules and parallel execution allow SEAS to model combat with decentralized control and autonomous agent action.

Studies have shown SEAS to be a competent platform to evaluate C4ISR effects. As mentioned previously, interest in capturing C4ISR effects has increased due to weapon systems dependence on ISR assets to shorten the decision cycle to

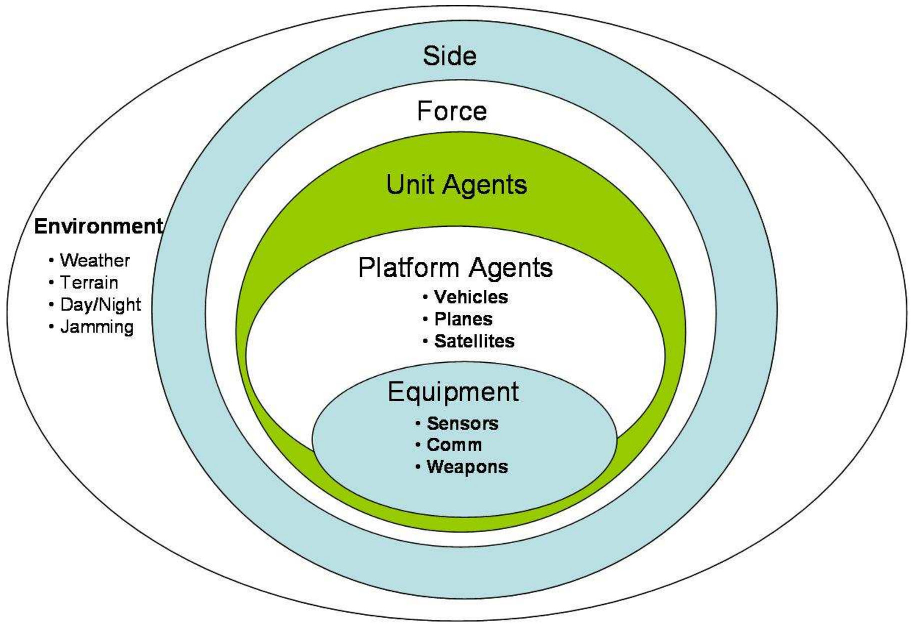

Figure 2.3 Diagram of Agents in SEAS

attack, maneuver, and exploit the enemy. If a new weapon system does not integrate well into a force C4ISR structure, its potential effectiveness is greatly reduced. The DoD has recognized this and called for an evaluation of any new weapons systems' interoperability within the C4ISR structure it is expected to perform in.

2.4 Architectures

Acquisition of any new DoD weapon system includes an integrated C4ISR architecture of that system. One of many definitions of an architecture is: "The structure of components, their relationships, and the principles and guidelines governing their design and evolution over time" [25]. The intent of the C4ISR architectures is to improve capabilities by quickly synthesising "go-to-war" requirements, aid efficient engineering, and rapid employment of improved operational system capabilities.

DoD and other Federal Agencies have developed architecture frameworks to provide rules and guidance for developing and presenting architecture descriptions in a uniform and consistent manner; ensure that architecture descriptions can be understood, compared, and related; define numerous products (graphical, textual, and tabular) to capture specific architectural descriptions [26].

A driving force for this wide usage of architectures in the DoD are the numerous mandates and policies imposed on Defense Acquisitions and Information Technologies. A well developed evolution and policy of the current Architectural Framework, Department of Defense Architectural Framework (DoDAF), to be used in DoD acquisitions can be found in Zinn [32]. In short these frameworks allow insight and foresight into issues like interoperability of a new weapon system into the military structure. Of course this requires integration into the C4ISR structure.

Information given from the architecture yields a static view of the systems interoperability. The architecture of the system is integrated to show the operational and system sides. This integration allows the DoDAF to provide a key element in the engineering design of a system, mapping operational activities to system functions in one architecture [32]. A brief overview of key components responsible for conveying this information is given.

The All-Views (AV) product encompasses some aspects of the three views discussed below. These are mainly high level executive summaries of the system that contain the following information: product definition, purpose, and detailed description. This discussion gives a brief overview of these products.

According to the DoDAF, the operational view (OV) is "a description of the tasks and activities, operational elements, and information exchanges required to accomplish DoD missions" [11]. There are seven products of the OV with the syntax OV-1 through OV-7. The detail of each product is not discussed here. The Systems View (SV) contains eleven products that describe the systems that support operations. These products are usually created in concert with or after the operational views. The Technical View (TV) provides general technical guidance for the OV and SV as well as technical and engineering standards for the SV. The diagram displays the interdependency of the views as evident in system views supporting operational activities. For more detail on the AV and the three views presented, reference Zinn or the DoDAF Volume II.

It is of interest to "view" the dynamic behavior of a system for evaluation purposes. Some of the products of the mentioned views give some *description* of the dynamic behavior. In particular OV-6 products describe the dynamic behavior of the system. According to the DoDAF: "The dynamic behavior referred to here concerns the timing and sequencing of events that capture operational behavior of a business process or mission thread for example" [8]. Some vehicle is needed to take these products and put them into motion. Petri Nets have been identified as a environment to create an executable model of these systems.

2.5 Dynamic Modeling of Architectures

In recognition of the static nature of architectures, there are several ongoing efforts to dynamically model systems presented via architecture. The dynamic modeling is of interest to evaluate if all the gears in the system turn, and how the system impacts the environment in which it is to be placed.

2.5.1 Petri Nets

Petri nets provide a modeling and simulation environment in which an executable model can be created from an architecture [13]. To understand how Petri nets can model dynamic behavior they first need to be defined. Next the petri models abilities and utility can be discussed.

A Petri Net is defined as a bipartite directed graph. Figure 2.4 displays the characteristic nature of the Petri net. Varying nomenclature for the petri nets may be used, our conventions follows that of Jensen [20]. We Define

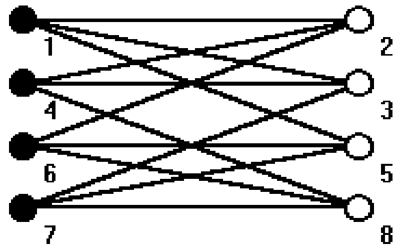

Figure 2.4 Example of a Bipartite Graph

$$
\begin{aligned}
PN &= < P, T, A, M > \\
P &= (p_1, p_2, ...p_n) \\
T &= (t_1, t_2, ...t_m) \\
A &= (a_p1, a_p2...a_pn)U(a_t1, a_t2....a_tm) \\
M &= (m_1, m_2, ...m_r) \\
\{n, m, r &\in I\} \ \ [I\ denotes\ the\ set\ of\ positive\ integers]
\end{aligned}
$$

where P is the set of places, T is the set of transitions, A is the set of directed arcs connecting places and transitions, and M is the set of tokens resident in places at a given moment.

Two distinct set of nodes exist. To be clear the arcs may only connect places to transitions and transitions to places. In the case of the petri net tokens are indistinguishable from one another. The number of tokens are depicted by a number within a dot, or a number of dots representing the number of tokens present. A non-negative number of tokens must be present in each place for a transition to fire.

Transitions are the crux of the petri nets ability to give dynamic qualities to a static system. As described by Handley [13]: a transition is enabled when each

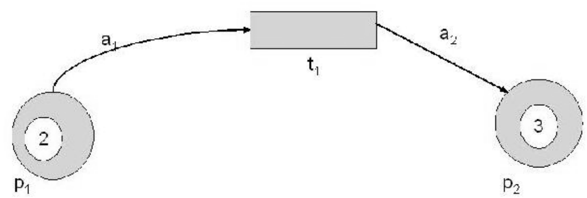

Figure 2.5 A simple Petri Net Example

input place to a given transition has at least one token. If this criteria is met, the transition may fire at this point and a token is taken from each input place to each output place. The new assignment of the number of tokens to each place is the dynamic characteristic of the system. When tokens are considered to be bits of information, real world systems can be modeled.

Colored Petri Nets (CPN) are a generalization of Petri Nets which define tokens as now having an attached data value called the token color. Restrictions are placed on which places tokens may reside (dependent on color). Transitions are enabled by a slightly different method. Input arch inscriptions specify the number and type of tokens that must be in place for a transition to fire. Tokens may be investigated and modified by the transition. Also, the transition's functionality may be represented by code that is run every time it fires [20].

The CPN is accepted as a rigorous method of dynamically modeling system activities. Levis and Wagenhals [22] have laid rigorous methods to produce an executable model via CPN. Levis and Wagenhals, et al generated an architecture (describing Mobile's SpeedPass "pay at the pump" system) and then used Petri Nets to produce an executable model. In the case of this research the architectures that are being considered are of new acquisitions in the DoD. It is important to note that the Petri-Net work done by Levis is of a contrived architecture made to fit the Petri-Net for a proof of concept. DoD architectures are not created with this in mind.

While the CPN is certainly a valuable tool for understanding the dynamic behavior of a system, it falls short in its ability to model a combat environment where the rules of engagement (ROE) are changing and the enemy model is learning and evolving. Clearly, to evaluate the military worth of a future system, you must do so in a simulated combat environment - one in which the enemy has his own architecture. The CPN is not well suited to accomplish this [31]. To address these concerns, one must look toward modeling and simulation

2.5.2 MITRE

The MIRTE corporation also has recognized the role of architectures and the standardization through DoDAF. The problem statement of a MIRTE study, Executable Architecture Methodology for Analysis (EAMA), states "static products lack means to conduct a proper dynamic analysis of IT systems capabilities, behavior and performance in its operational environment over time" [26]. The following solution has been proposed. Develop a methodology to convert static architecture products into executable architectures to support dynamic analysis of a system or capability, and measure process performance along with organizational work efforts and resource utilization over time. Develop a federation of simulations that represent the mission threads (business processes), communications networks, and operational environment for the system being analyzed. Another goal is to generalize the methodology to work with multiple models and multiple modeling tools.

EAMA intends to extract Measures of Effectiveness (MOEs) from the architecture using Army combat simulation EAGLE, and the commercially available business process model BONAPART. BONAPART, a Petri-Net tool, evaluates actions to be taken at each time step, and these are fed into EAGLE for execution while the state of the system is updated after any executions. This process allows the system represented by the architecture to be embedded within a combat model, EAGLE, for evaluation. While the intermediate business process model step in this approach will not be duplicated the overall goals and methodology parallel those of this study. This ongoing research effort has not yet produced an evaluation of a system, but recognizes the potential of architectures can be reached through the help of modeling and simulation.

2.5.3 RAND

In an effort to improve the C4ISR capability SEAS version 2.3, the RAND corporation compared a baseline C4ISR architecture, an improved airborne architecture (IAA), and a satellite wide area surveillance architecture (SWAS). The C4ISR architectures were not created from DoDAF products, but from various Air Force and Joint Publications documents [29]. The activities of the AOC are complemented by two agents developed for the study. The first agent is a collection planning agent to prepare a detailed plan for requirements and time in which the ISR assets collect information. The second agent, ISR battle manager, dynamically determines how to carry out the collection plan. While the architectures represented in this study are slightly out of date, the level of effort required to accurately model the processes of the AOC is of interest.

In addition to the added agents this study also enhanced sensors and ISR metrics. These changes have been incorporated in to SEAS version 3.2 which is the version used in this study. Further discussion of the finding of this study will be discussed in chapter 5 as they impact the capability of SEAS to represent C4ISR systems.

2.6 Summary

Recently there has been a significant trend in the utilization of ABM within the DoD in efforts to capture C4ISR effects. The AFSAT has incorporated SEAS at the mission level, and SEAS has been used for analysis in a wide range of scenarios. Typically, SEAS analysis is used to identify first-order non-linear C4ISR effects for further investigation. C4ISR interoperability issues have led to requirements for all new major weapon systems to be placed within a DoDAF. The completeness of the DoDAF increases as the acquisition cycle progresses to eventually include dynamic views which are essential in creating an executable model of the system. The executable model is to provide an environment for the system to be evaluated in over time. Evaluations of the system by the acquisition and warfighting communities would benifit from a transfer of information from a system's architectural products into SEAS for further analysis of military worth in specific conflict scenarios.

3. Methodology

3.1 Overview

This research is accomplished in three parts. First, the transfer of data from a Time Critical Target (TCT) architecture in DoDAF format to SEAS will be investigated and accomplished. Second, once the data is transferred to SEAS a MUA of the TCT process executed by the AOC represented by the architecture is conducted. Finally, insight of how the AOC may perform its mission more effectively based on MOE(s) from the MUA will be passed back to the architecture.

The gathering of data from the architecture requires detailed knowledge of the available architectural products. Collaboration with AFIT System Engineering Student Captain Drew Zinn has provided insight and data products of the proposed architecture. The architecture chosen for evaluation and translation into SEAS is of an Aerospace Operation Center (AOC) created for the Air Force by the MITRE Corporation in Hampton, VA. The provided architecture documents seven operational activities to be accomplished by the AOC. The operational activities are:

CAOC-1.0-Produce Joint Air Operations Plan
CAOC-2.0-Provide Specific Guidance
CAOC-3.0-Task Available Capabilities/Forces
CAOC-4.0-Manage Aerospace Operations
CAOC-5.0-Evaluate Results of Joint Aerospace Operations
CAOC-6.0-Perform Airspace Control Authority (ACA)
CAOC-7.0-Plan, Task, Execute, and Assess Theater Air

The objective was to utilize architectural products to build a SEAS simulation to capture these operational activities. In previous work in creating executable models from DoDAF architectural products, authors Levis and Wagenhals [22] emphasized the importance of a dynamics model in the process. The dynamics model is represented in the DoDAF with the OV-6. However, the OV-6 has not yet been created for the AOC's architecture.

Often when an area of architecture requires a higher level of attention a subset of activities are modeled. This subset of activities which are modeled at a higher level of detail than the rest of the architecture is called a "key thread" [32]. Derived from the AOC architecture, a Time Critical Targeting (TCT) key thread is also available. The TCT is available as its own architecture which includes an OV-6 product. The TCT and AOC architectures share much of the same data, were built by the same modeler, and describe common systems (although at different levels of detail). Therefore, the goal of transforming the static views of architecture to a dynamic executable model remains the same, but information is now extracted from two architectures. Furthermore, the objective of the study has transformed into capturing the operational activities of TCT conducted by the AOC.

TCT-1.0 Analyze ATO period for dynamic targeting opportunities
TCT-2.0 Monitor battlespace for dynamic events
TCT-3.0 Verify event is/is not of interest
TCT-4.0 Adjust Theater ISR to support dynamic air operations
TCT-5.0 Define target/target set
TCT-6.0 Determine target significance/urgency
TCT-7.0 Validate target/target set
TCT-8.0 Nominate engagement options
TCT-9.0 Execute engagement options
TCT-10.0 Attack target
TCT-11.0 Conduct dynamic assessment of target

Utilizing the aforemention OV product this research evaluates the AOC and TCT architectures to ensure we properly model the TCT thread within a SEAS warfile. The warfile is a text file describing scenario agents along with their heirarchy and behaviors. Preceding these actions an evaluation of the communications process, command and control, and execution of orders in SEAS is conducted. This allows for proper placement (logical and spatial) of the assets, communications, and physical characteristics of the system represented by the architecture. The means and validity of transferring each of these categories of data will also be addressed in this chapter. With proper characterization of all TCT aspects in SEAS we have an executable model for the architecture. It is important to note that the coding of the AOC agent and TCT activities in SEAS, not only changes the AOC asset in the scenario, but changes the entire model of war. The actions of AOC and other agents are altered with the change and creation of agent orders. Thus, the change of interactions between agents create a new model for the agents to follow.

A MUA is conducted within a probable environment of future U.S. engagements. Traditional and non-traditional MOEs are used to capture the effect of TCT in the scenario chosen. Any adjustment in activities of the TCT to improve the MOE(s) will then be recommended as added capabilities or transformation of operations of the AOC. These recommendations will be passed to Zinn and attempted to be captured in the architecture.

3.2 Scenario

A number of priority operational challenges have been identified by a capabilities-based planning study prepared for the Office of the Secretary of Defense [7]. One operational challenge has been identified as an effective stop-the-killing intervention in a small-scale contingency. A recent encounter with this type of scenario is the U.S. involvement in Kosovo in 1999. Insight to our capability to be effective in this type of operational challenge can be accomplished via the agent based combat simulation SEAS.

The Space and Missile Center Transformation Directorate (SMC/TD) has created a warfile in SEAS to represent a typical mission in the Kosovo war. The warfile created in SEAS contains all information of the environment, entities, and timing of the events to be modeled. This particular warfile consists of U.S. forces, Serbian forces, and Kosovar militia and civilians. In SEAS the U.S. forces are owned by a master unit agent, the AOC, which passes orders to its subordinate assets. The AOC's available assets are those representatives of what we would bring to war in a smaller scale theater conflict such as Kosovo. Figure 3.1 represents the assets of the USAFE (United States Air Forces Europe) in the scenario. The Unit agents shown are all owned by the USAF Combined Aerospace Operations Center (CAOC) agent. The platform agents are encompassed by the unit agents as they are subordinate agents. Both the unit and platform agents own equipment. All these agents act within the environment which has an effect on movement and individual sensors probability of detection.

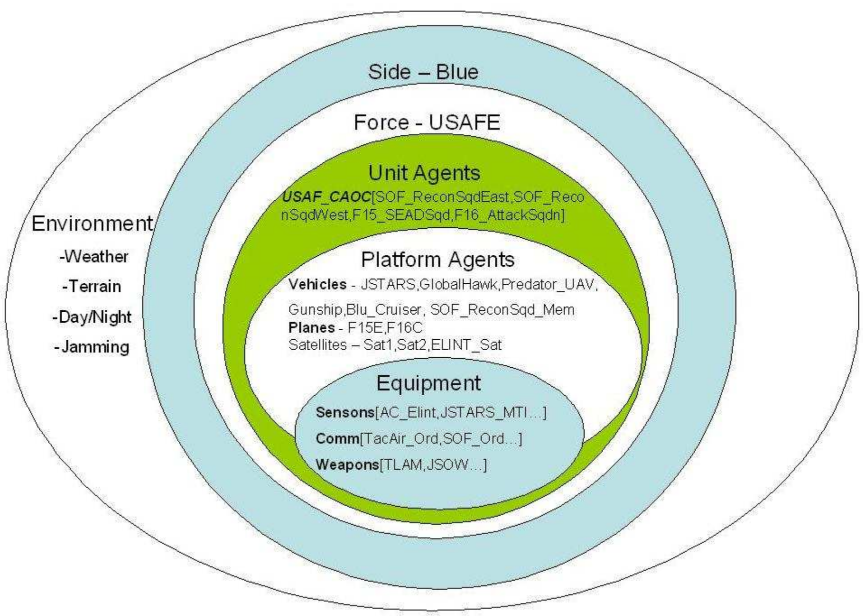

Figure 3.1 Blue Agent Force Structure

In contrast, the AOC architecture was built considering a current theater response package (TRP). The AOC architecture provides information on all elements common to most AOCs put together for a TRP. Available external assets such as long range bombers and reach-back capabilities of the AOC exceed that of the SMC scenario. No new assets have been modeled in SEAS as the goal is to capture the activities of TCT utilizing the assets available in the Kosovo scenario provided.

The provided scenario representation of Serbian forces remain unchanged. Serbian unit agents include air defenses, ground targets, and three army divisions. These are labelled as *Serb _Pristine _AD*, *Serb _ groundtgts*, and *Serb _Armor1,2 &3* respectively in the warfile. The Serbian air defense unit consists of two surface-to-air(SA)-6, one SA-3, and one air traffic control unit agent. The air defense unit owns a tactical communication system, *RedTac _Ord*, to relay commands and broadcast variables. Each SA-6 battery consists of two radar vans, *RedSA61RadarVan*, eight transporter erector launchers(tels),*RedSA61Tel*, and the *RedTac _Ord* communication channel. Once the agents and communication equipment are created in SEAS a hierarchal display of the force structure is available. Figure 3.2 displays the breakdown of the Serbian force agent to the platform agent level. Each platforms owned sensors, communication equipment, and weapons are also shown.

The Serbian force is not centralized as is the blue force possessing the CAOC unit agent which owns all other blue agents. Of course, the five main Serbian unit agents may pass orders to their subordinate agents, and they all may share information as they are connected via the *RedTac _Ord* communication equipment. All units are given orders to operate as expected in war. For instance, the SA radar vans are given orders to hide when information is passed that an F15 is near, or to hide and move after firing a missile.

As in the Serbian forces no regular forces are available for the Kosovars. As seen in Figure 3.3 only scattered poorly communicating militia and civilians are modeled. Cellular phones, bells, and shouting are the forms of communication for

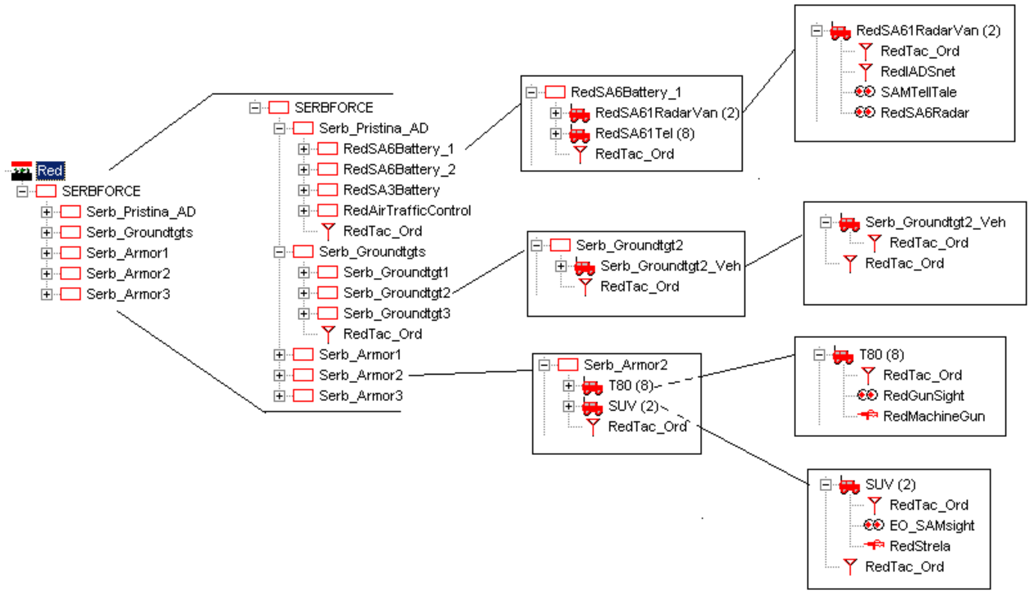

Figure 3.2 Serbian Force Structure

the Kosovars. Vehicle platform agents are bicycles, pushcarts, and tractors. The only sensors are the human eyes, or *K-eyes* as in the warfile. This irregular structure and equipment are able to be modeled due to the flexible nature of the SEAS warfile. Instead of the Kosovars being placed in aggregated masses at certain locations they can be modeled as agents who can pass along information to the U.S. forces and hide from the enemy. This adds elements to the war scenario that are difficult and not typically modeled in other simulations.

The orders and communications of the Serbian and Kosovar forces will not be altered. They are evaluated to provide knowledge of the enemies capabilities and possible behaviors to illuminate weakness for TCT to take advantage of.

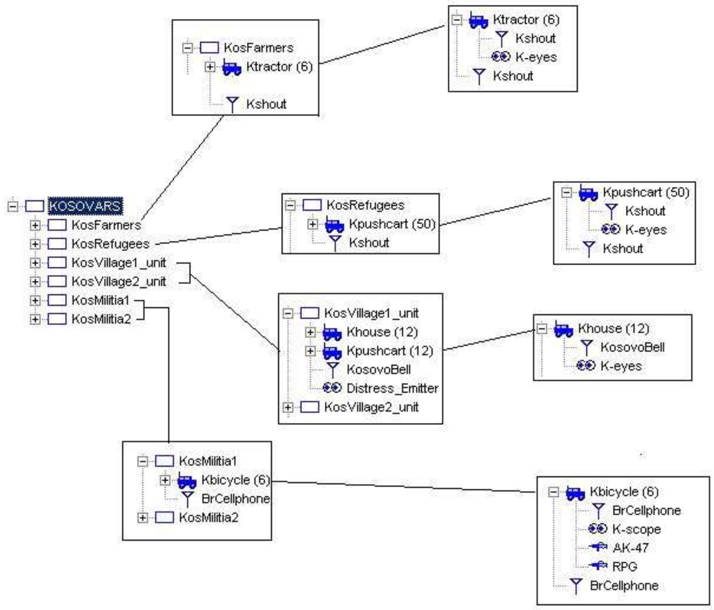

Figure 3.3 Kosovar Force Structure

3.3 Data Transfer to SEAS

3.3.1 Communications

The first issue addressed in the transfer of data from the architecture to SEAS was the communication structure of the AOC with its assets in carrying out the TCT mission. Modeling the information flow between assets is critical to getting an accurate picture of the C4ISR effects from the simulation. If a satellite is passing information to the AOC, but the AOC is not able to transmit that information to the proper assets the benefit of the satellites information is not captured.

3.3.2 Communications in SEAS

In SEAS communication equipment is explicitly assigned to each agent. Communication equipment have definable range, delay, and reliability of the messages they transmit and receive. The messages pass over channels which have a definable time that a message will be held, and a maximum number of messages that can be passed over the channel in one time step. Weather and terrain effect the reliability of message transmission and range of message reception respectively.

The mode and message type attributes of the communication equipment define the direction and type of information to be passed respectively. The mode attribute establishes the direction of the message flow by:

$$Mode = 1, \; transmit$$
$$Mode = 2, \; receive$$
$$Mode = 3, \; both$$

SEAS allows three message types to be passed over the communication network.

$$MessageType = 1, \; target\ sightings\ are\ passed$$
$$MessageType = 2, \; commands\ are\ passed$$
$$MessageType = 3, \; broadcast\ variables\ are\ passed$$

Target sightings are passed as locations in (latitude(lat),longitude(long), and altitude(alt)) format. Commands tell the receiving agent to move to a new location, abort a mission, or any numerous other commands the analyst programs. More than one message type may be passed over communication channel(s) by adding the number of the message types to be passed. For instance, a five placed in the message type attribute would allow commands and broadcast variables to be passed over the communication channel.

To aid in establishing understanding of the current SEAS scenario's communication the network in Figure 3.4 was created. The nodes represent the AOC master unit (center), and all platform agents subordinate to the AOC. The connecting lines represent the communication links between all agents. The nomenclature of each communication link conveys the communication device name used in SEAS to connect the nodes. The message type and mode are represented in each link. For instance, the link between the Predator and the AOC is labelled *PredUAV _ TAC _ ORD(2,3)*. Thus allowing commands (Message Type 2) to be passed over this link in both directions (Mode 3). Any arrow pointing directly into a platform or unit agent represents sensors owned by the agent feeding information to the platform. A separate network, Figure 3.5, has been built for the satellite communication with the AOC. Satellites are not controlled by the AOC or any other agent. They transmit information to the AOC and the Special Operations Forces (SOF) unit agents in this scenario.

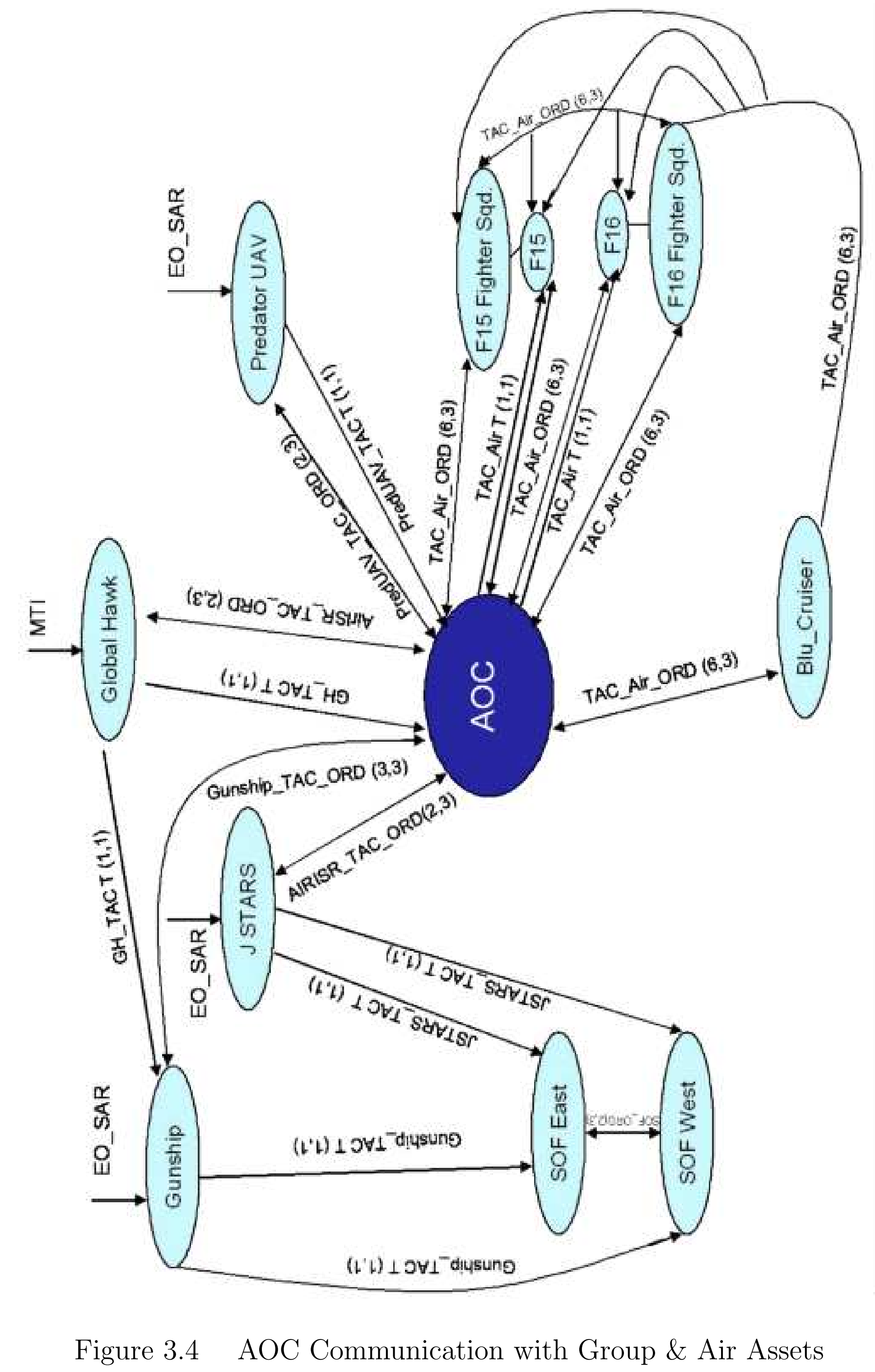

Figure 3.4 AOC Communication with Group & Air Assets

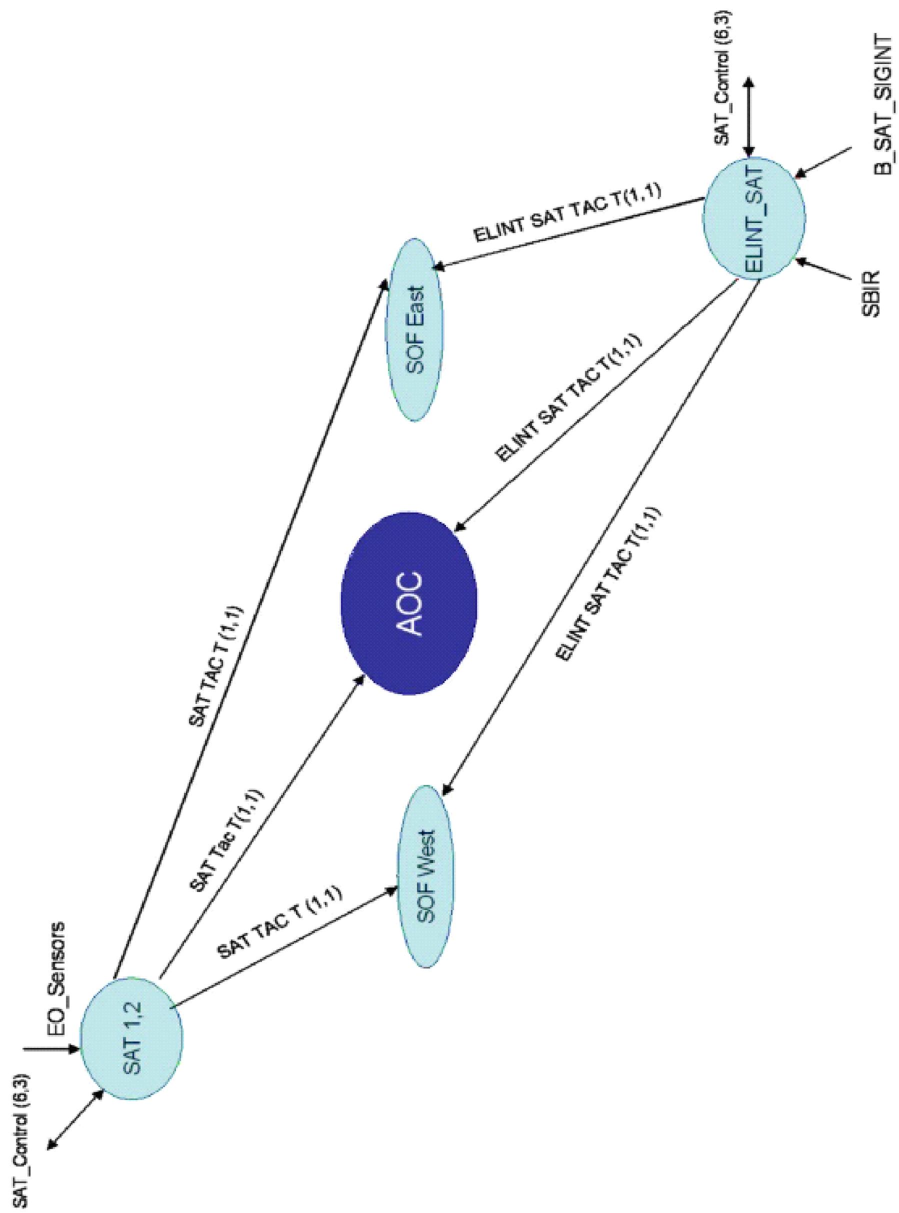

Figure 3.5 AOC & Satellite Communication

The networks were created after the warfile, but the intent is to create a method to keep track of all communication links between agents. This became instrumental when sorting through architecture products, shown in the next section, to keep track of communication information.

3.3.3 Communications in DoDAF

The DoDAF offers most communications information in the architectural views OV-3 (Information Exchange Matrix) and SV-2 (Systems Communication). An OV-3 provides all information exchanges between nodes in the architecture, and even the information within a node. Within the AOC node there are numerous instances of information passed from one cell to another cell to accomplish activities. The information exchange gives insight of the message type and mode that is established between agents in SEAS.

A need line tells who is passing the information and who is receiving. An example of the provided OV-3 product from Zinn displaying all the AOC to fighter/bomber platforms information exchanges can be seen in Figure 3.6. At a minimum this requires that communication is established between the two agents in SEAS. The title of the information exchange and the sending and receiving activity blocks give insight about the information to be passed. An information exchange block labelled mission change orders conveys that orders, message type 2, will need to be passed over the communication channel in SEAS. This is reaffirmed by the names of the sending and receiving activities (i.e. divert, cancel, and re-role mission are required). The information exchange matrix was then examined for need lines from fighter/bomber to AOC. This is done to establish the mode to be set on the agents communications.

The OV-3 provided is that of the broad scoped AOC architecture, so only those need lines involving assets available in the Kosovo scenario have been examined. The OV-3 does not provide the means that the information is to be transmitted by, how often, or its precedence over other information. The SV-6, system data exchange matrix, does include information on media, format, protocol, and size of data which would answer some of these questions. However, the information in these columns is not filled in, so we are left to ascertain these characteristics from other products. It is preferable to use the OV-3, at any rate, as explained in Zinn's thesis, section

NEED LINE	INFORMATION EXCHANGE	SENDING NODE	SENDING ACTIVITY	RECEIVING NODE	RECEIVING ACTIVITY
CAOC to Fighter/Bomber	Operations Direction & Guidance	CAOC	CAOC-4.0-Manage Aerospace Operations Execution CAOC-4.5-Manage dynamic targeting execution CAOC-4.5.3-Manage unique dynamic target missions	Ftr/Bmbr	Ftr/Bmbr-2-Receive C2 information
CAOC to Fighter/Bomber	Mission Change Orders	CAOC	CAOC-4.4.4.2.2-Divert, cancel, re-role missions as required CAOC-4.6.4.2.3.1-Direct diverts as required	Ftr/Bmbr	Ftr/Bmbr-2-Receive C2 information
CAOC to Fighter/Bomber	Dynamic Targeting Execution Orders	CAOC	CAOC-4.5.2.9-Execute engagement option (Engage) CAOC-4.5.2.9.2-Issue execution orders	Ftr/Bmbr	Ftr/Bmbr-2-Receive C2 information
CAOC to Fighter/Bomber	Critical aircrew information	CAOC	CAOC-7.3.4-Pass critical information to aircrews for time	Ftr/Bmbr	Ftr/Bmbr-2-Receive C2 information
CAOC to Fighter/Bomber	On-call CAS Scramble/ Execution Order	CAOC	CAOC-4.5.3.1.1-Manage sortie diversion for on-call CAS	Ftr/Bmbr	Ftr/Bmbr-2-Receive C2 information
CAOC to Fighter/Bomber	CSAR Direction and Guidance	CAOC	CAOC-4.5.3.2-Coordinate CSAR missions (assumes existence of an JRCC in the CAOC-4.5.3.2.5-Execute engagement option (Engage) CAOC-4.5.3.2.5.6-Coordinate	Ftr/Bmbr	Ftr/Bmbr-2-Receive C2 information
CAOC to Fighter/Bomber	CSAR Execution Order	CAOC	CAOC-4.5.3.2.4-Determine engagement option (Target) CAOC-4.5.3.2.4.2-Receive approval for engagement option	Ftr/Bmbr	Ftr/Bmbr-2-Receive C2 information
CAOC to Fighter/Bomber	CSAR Engagement Status Update	CAOC	CAOC-4.5.3.2.5-Execute engagement option (Engage) CAOC-4.5.3.2.5.6-Coordinate intel and SERE debrief of survivor	Ftr/Bmbr	Ftr/Bmbr-2-Receive C2 information
CAOC to Fighter/Bomber	CSAR Asset Retasking	CAOC	CAOC-4.5.3.2.5.4.2-Retask assets as required	Ftr/Bmbr	Ftr/Bmbr-2-Receive C2 information

Figure 3.6 Example OV-3 Product - AOC to Fighter/Bomber

4.1.2. Regardless whether the OV-3 or SV-6 is used to describe the content of the data, neither product is intended to describe how it is transmitted.

To do this, the SV-2 "System Communication Description" is required. The SV-2(see Figure 3.7) may provide other information such as bandwidth, frequency, and the communication system used. This information aids in the reliability, delay, and range parameters set in SEAS. However, we are not dealing with one complete architecture and the SV-2 is provided via the AOC architecture. The SV-2 for the AOC architecture is incomplete leaving much information to be extrapolated from the OV-3. Most of the information can be discerned from investigation into the receiving and sending activities, but the incomplete architecture creates the opportunity to misrepresent the system.

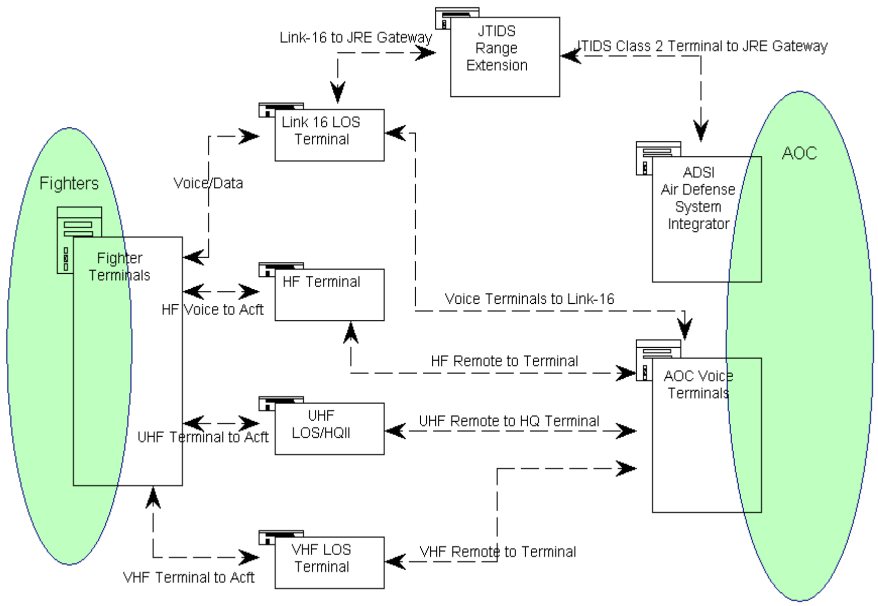

Figure 3.7 Example SV-2 product - Systems for AOC to Fighter Bomber Exchange

The sending and receiving activities shown in the information exchange matrix can be further investigated in the activity model diagrams which are products of the OV-5. The TCT and AOC architectures provide OV-5 products.

An activity is represented by IDEFØ, a method designed to model the decisions, actions, and activities of an organization or system, which is a simple graphical language of box and arrows (see Figure 3.8). The input to the activity is information received from another activity or source. Controls are constraints such as the Law of Armed Conflict (LOAC) that govern our conduct in war. Mechanisms are who or what conducts the activity. In this case the who may be a C2 warrior in the AOC, and the what may be a linear programming model to optimize received inputs. Outputs are the data product that are passed on to the next activity.

Arrows flowing in and out of the activity blocks represent the inputs, controls, outputs, and mechanisms (ICOMs) used to guide the activity's actions and decisions. The ICOMs are labelled with the information it passes. This allows us to focus on the

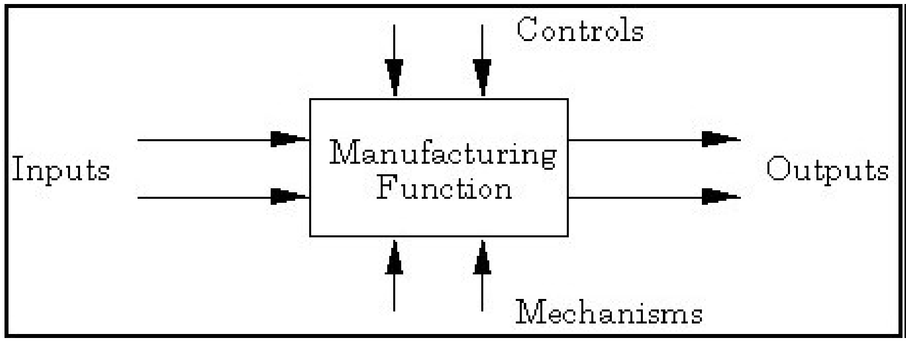

Figure 3.8 IDEFØ Format

information exchanges between the activities in TCT via the TCT's OV-5 products. These are derived from the same information exchanges described in the OV-3 from the AOCs architecture. The OV-5 product offers a definition of the information exchange of each ICOM arrow. Instead of relying only upon the information exchange and sending and receiving activities titles from the OV-3, the definition adds to the insight of the information to be passed over the communication link. In the case of TCT and AOC architectures few ICOMs are completed with these definitions.

At this stage of the analysis it is determined that the communication structure built by SMC provides the necessary communication channels for platforms to share information and send/receive orders. Only a few changes to the message types have been made to the original structure. This is in part due to lack of information provided via the architecture. This shortcoming will be discussed in Chapter 5. In short, the OV-3 and OV-5 provide the information to be exchanged and the SV-2 shows communication systems to be used, but which information flows over individual channels between two nodes is not available. All information flowing between two agents has been aggregated into potentially fewer channels between agents.

3.3.4 Activities

With communications and the correct hierarchy of agents complete for the blue force, information involving activities necessary in conducting TCT was then sought from the architecture. TCT targets are those that fall below the joint integrated prioritized target list (JIPTL), or desired information is received to late to add to the Air Tasking Order (ATO), a scheduled attack plan for the AOC's assets to execute. In a typical TRP TCT targets are attacked with a small subset of the AOC's assets which are set aside explicitly for this purpose. However, due to the nature of the engagement the AOC modeled in the Kosovo baseline scenario does not produce or follow an ATO. Therefore, this study considers many of assets owned by the AOC agent as available and tasks them to execute the TCT mission.

The TCT architecture OV-5 product yields a diagram in IDEFØ of the eleven operational activities, stated in the overview. Figure 3.9 displays TCT operational activities 1 and 2 with their associated ICOM arrows. This product is somewhat useful in determining the type of orders needed to accomplish these activities in SEAS. However, the OV-5 IDEFØ format does not capture sequence or timing of the events that generate these activities.

In the TCT architecture each OV-5 has an OV-6a rules model associated with it. The logic or timing associated with activities are captured via the IDEF3, process flow and object state description capture method, used in the OV-6a. Logical statements with nomenclature *X*, *O*, and *&* (XOR,OR,AND) are utilized to show the information needed to move along within the activities processes.

In Figure 3.10 a breakout of the OV-6a rules model for the Attack Target activity is shown. The large blocks describe units of behaviors to be executed once information is passed along to it. In this case if the decision is made that the target is static then that target is to be monitored for movement. If no movement is seen, the target coordinates are then passed along to an *O* (OR) junction where the target is updated with coordinates rather than a vector.

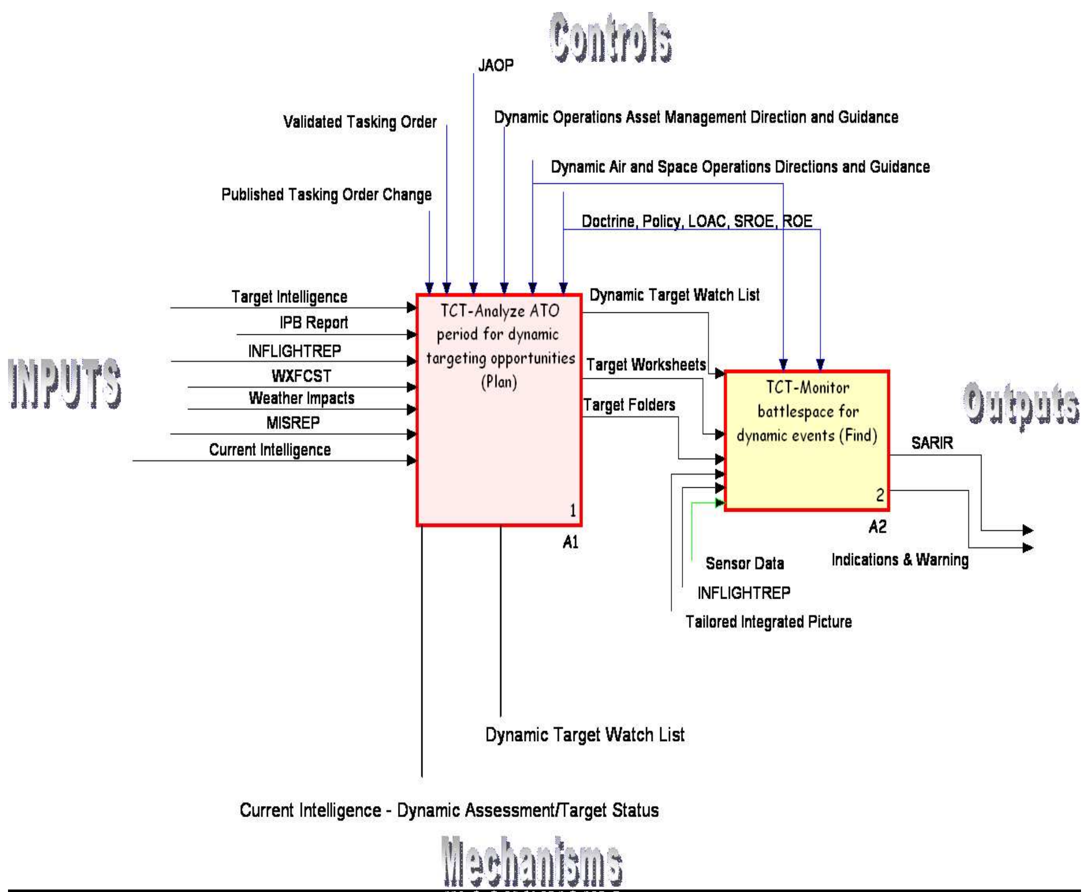

Figure 3.9 Activities 1 & 2 from the TCT OV-5 product

To model all OV-6 unit of behaviors and decisions nodes in SEAS would require the creation of a vast number of agents and variables to manage data and process decisions. The information passed through the TCT is not solely generated from the activities of the TCTs OV-5, but from all the AOC activities where no OV-6 is available. In the event the data becomes available it is possible to create all these agents and variables, but to model the hundreds or potentially thousands of agents in an AOC shifts from the mission/campaign level nature of SEAS. SEAS is generally used to identify first-order C4ISR effects for further investigation via other means.

Instead a synthesis of OV-5 and OV-6 products was created to model each activity such that the model captures the essence of each activity. Critical events and decisions completed by each activity were mapped to internal SEAS functionality

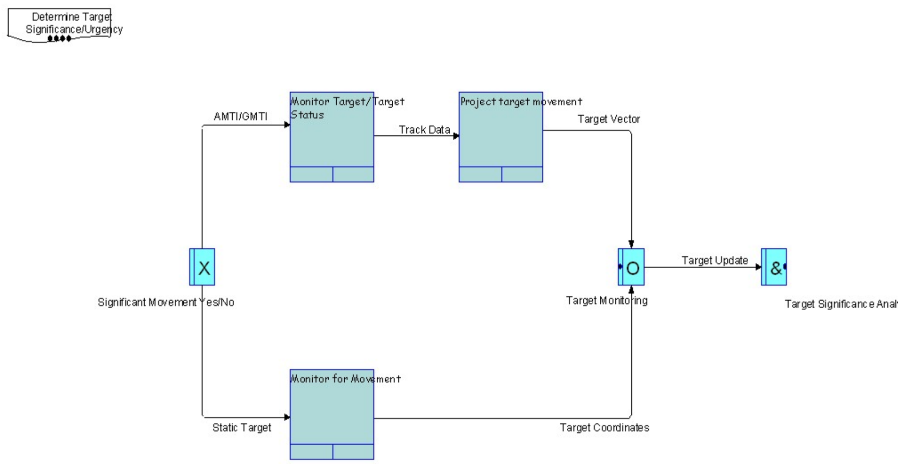

Figure 3.10 Breakout of Rules Model: Attack Target Activity

or code in the agent orders written. As an aid to write the agent orders from these products Zinn provided psuedocode for each activity. The activity was written in an *IF, Then, Else* format translating the graphic product of the OV-6 and providing language relevant to most standard programming languages. Below is an example of the psuedocode for activity 7.

Pseudo Code for Activity 7 - Based from IDEF3 Diagram

Determine if the target is valid

IFthe target is on the (Dynamic Target Watch List) OR (the ATO/JIPTL)

Then continue validation

Else

IF(The target is consistent with Dynamic Target Execution

direction and guidance for the JFAC) AND (LOAC and ROE have been reviewed)

Then continue validation

Else not a TCT-pass on to ATO planners

Considerations: Predict adversary reaction

Provide collateral damage consideration

Determine political sensitivity of target

OUTPUT: Nominate Validated Target for Attack

Figure 3.11 is a decision flow of the activities in the TCT created from the OV-5, OV-6, pseudocode products. Many underlying behaviors of the OV-6 are not displayed, but essential to the overall process flow shown. This diagram serves as a guide to code and evaluate SEAS such that each of the following activities are explicitly or implicitly accounted for in the scenario.

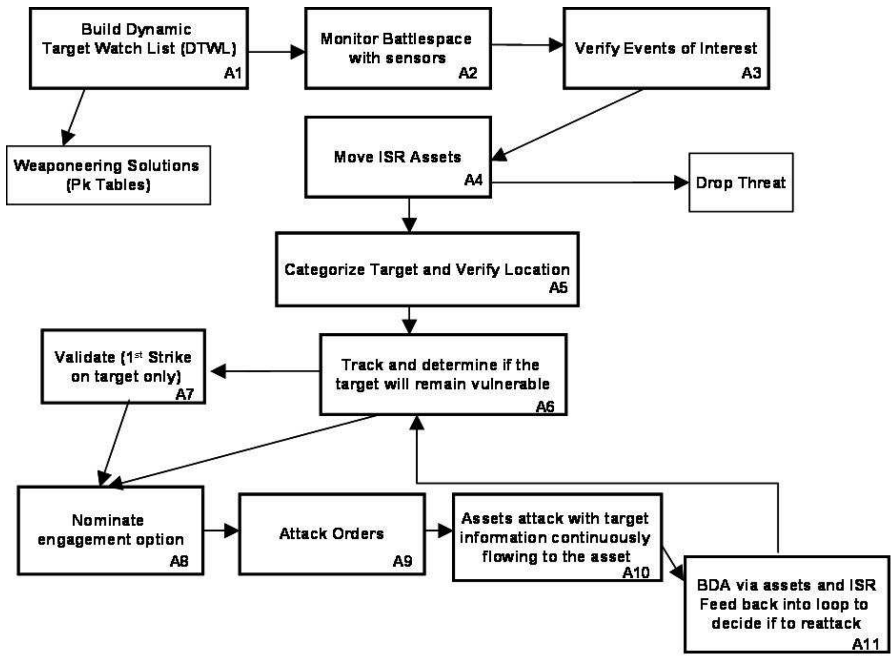

Figure 3.11 TCT Process Flow Diagram Created via OV-5 & OV-6 Products

The first activity, labelled *A1* in the TCT flow diagram, is the generation of a dynamic target watch list (DTWL). This is a prioritized list of potential enemy targets expected to be seen in the scenario. From the DTWL this activity produces the weaponeering solutions for possible targets in the battlespace which are to be monitored. The weaponeering solutions are defined by the probability of kill (P_k) table in SEAS. For instance, a Joint Standoff Weapon *JSOW* carried by the F15, is assigned a .8 probability of kill P_k to the *RedSA61RadarVan*. SEAS does not allow a weapon to engage a target if no P_k assignment has been made. The creation of the P_k table is left to the analyst as targets and weapons are scenario dependent.

These weapon to target pairing are predetermined before allocating assets for TCT missions.

The DTWL representation is completed in two parts. First, an entry is made in the probability of detection (P_d) table in SEAS assigning sensor detection for a particular target type. This follows the logic of the P_k table, with no detection possible if no assignment has been made. Once the assignment is made it is then possible for the sensors to pass information among agents via the communications network. Second, following the DTWL priority hierarchy set forth by decision makers in the AOC, an array of global variables representing each target type is defined below. This array explicitly categorizes each target type which allows the AOC agent to make decisions such as which target to attack first and priority of the retasking of ISR assets.

F15_DTWL[0] = "RedSA3RadarVan"

F15_DTWL[1] = "RedSA61RadarVan"

F15_DTWL[2] = "RedSA62RadarVan"

F15_DTWL[4] = "RedSA61Tel"

F15_DTWL[5] = "RedSA62Tel"

F15_DTWL[6] = "T80"

F15_DTWL[7] = "Serb_Groundtgt3_Veh"

F15_DTWL[8] = "Serb_Groundtgt2_Veh"

F15_DTWL[9] = "Serb_Groundtgt1_Veh"

F16_DTWL[0] = "Serb_Groundtgt1_Veh"

F16_DTWL[1] = "Serb_Groundtgt2_Veh"

F16_DTWL[2] = "Serb_Groundtgt3_Veh"

F16_DTWL[3] = "T80"

Activity 2, monitor the battlespace with sensors, is left to the inherent abilities of SEAS. Once an agent is deployed its sensors have the ability to detect targets on its P_d list. In the scenario ISR assets are placed in orbits above the battlespace.

Activity 3, verify the events of interest, is accomplished implicitly by the use of sensors probability of identification (PI_d) attribute. PI_d is the contribution the sensor provides in correctly identifying the target where 0 denotes no contribution. SEAS requires a PI_d_Commit threshold to be met before a weapon may attack a target. This is to avoid attacking targets where poor or little data is available. The AOC is privy to all sensor data collected from its agents, so when a target is sighted even with a low PI_d, the AOC is aware of the information. Once a target is detected it is placed on the sensors platform agents Local Target List (LTL) and the information is sent to the AOC. The information passed includes Target Location Error (TLE), Target Velocity Error (TLV), and mass of the target. If the information passed is categorized as a threat on the DTWL further action may be warranted.

Activity 4 is to Move ISR Assets. In the baseline scenario a Global Hawk flew an orbit on the edge of the main area of interest. Due to the Global Hawk's position and sensor capability it has been designated as the reassignable ISR asset.

The Global Hawk may be programmed in a variety of ways in the TPL to move to a new TAO. In the orders displayed below the Global Hawk ISR asset is told to move once sighting a target on the (DTWL), *Priority_Mass 0*, is sensed from a blue asset passing back a probability of identification (PI_d) lower than the PId_Commit threshold. The Global Hawk moves to a new TAO, *Investigate_TAO*, with the target sighting location as the center. The Global Hawk completes at least one orbit before it can be told to move to another location, and remains in that orbit as long as it is sensing targets, not in danger, or no other higher priorities exist.

```
Orders
    Declare Global Priority_loc Priority_mass
    Declare Global Investigate_TAO
While me -> Status! =2
End While
While (me->Status ==2)
If Priority_mass > 0
    Move Investigate_TAO
  While me -> Location! = me -> Goal
  EndWhile
Else distance(Priority_loc, Location(19.484556, 42.73121331, 4000)) < 10
     Move "GH_Oribit"
EndIf
EndWhile
EndOrders
END
```

Activity 5, categorization and verification of the target, is accomplished while this Global Hawk is flying its new TAO when sighted sensor information (TLE, TVE, and mass) is placed on the Global Hawks LTL. The LTL is passed back to the AOC at some defined broadcast interval (BI). Now the AOC categorizes the target type and location with a higher probability of the $PI_d \geq PI_d\_Commit$.

Activity 6, to track and determine if the target will remain vulnerable, can be accomplished in SEAS, and is target dependent. TLV information from the sensor(s) detecting the target is available. Once a target is placed on a platforms Local Orders List (LOL) the target distance is calculated via the TLV, time of the sighting and current simulation time. If the target is expected to be out of the weapons range (defined in the weapons section of the warfile) the weapon is not fired. Also, rules may be written to determine a targets' vulnerability. Within this scenario we are presented with the case of mobile radar, tanks, and trucks that do not remain vulnerable as they can hide via movement and/or no sensor emissions. No solution to this activity was coded in the study.

Activity 7, validation of target in the case of a first strike, is left to the PI_d and PI_d_commit measures. Omitting Activity 7 after the first strike is noteworthy as it can not be determined from the OV5 product alone.

Activity 8, nominate engagement option, is accomplished in part by P_k tables, and in part by orders. The P_k assignment allows platform weapons to be launched against the target. F15s are assigned to all scenario targets, and F16s are assigned to tanks and ground targets (see the DTWL). When assets are available they fly to the highest priority target type first. Priority has been set in the F15 and F16 squadron unit orders.

Activity 9, attack orders, are often scenario and target dependent. Each plane agent has orders to avoid the nearest SA radar as it moves towards a target. Other orders such as flying through various waypoints, directly flying to a target, number of munitions and planes sent to attack, or turning off radar on the way to a target are some possible attack orders. This is a very flexible and easily manipulated functionality of SEAS. For this scenario basic orders for the plane agents have been established based on the given threats.

Activity 10 is to perform BDA via asset and ISR sensors and then decide if to reattack. Each sensor has a BDA probability and time associated with it. If a target is BDA as dead, it is removed from the LTL and is no longer attacked. If the BDA is not accomplished or shows no damage, the target remains on the LTL and the platform may be assigned orders to reattack. SEAS does not model the case of considering a live target dead.

3.3.5 General Attributes

Communications and orders encompass much of the necessary information to define an agent and its behavior in SEAS. However, they do not account for attributes such as performance characteristics (i.e. speed), number of personnel associated with the agent, and deployment procedures. Architecture data products used to fill in these general attributes are the SV-7, OV-4, and SV-1. For this study most general attributes have been set from external sources. Zinn [32] discusses the deficiencies in the architecture data products to generate this information.

3.4 Verification and Validation

Several tools and techniques have been employed throughout the process of building the SEAS model of AOC's TCT activities to ensure a valid model which captures the events and behaviors of interest. Some standard methods employed were a structured walk-through of the code, consultation with experts, viewing the animation, and looking for reasonable output. SEAS, provides a details and debug window that were useful in this process.

The following states some of the benefits of the mentioned model verification and validation methods. A structured walk-through of the code was accomplished every time an agents orders had been changed. Each agents orders are dependent upon global and local variables. If these variables are not updated with a change in orders some action(s) may not occur. Consultation with experts at SMC (a primary user of SEAS), Sparta Inc. (model managers), and RAND (analysts) were also used throughout this study. These analysts provided insight to code limitations, past studies, and a second set of eyes on the code generated from this study. Viewing the animation was integral in determining if the global hawk moved to a new TAO. The screen capture (see Figure 3.12) of SEAS shows TAOs in white. The blue sensor circles represent the agent senor's field of regard. It is clear to see the global hawk as diverted from its TAO to investigate a potential target.

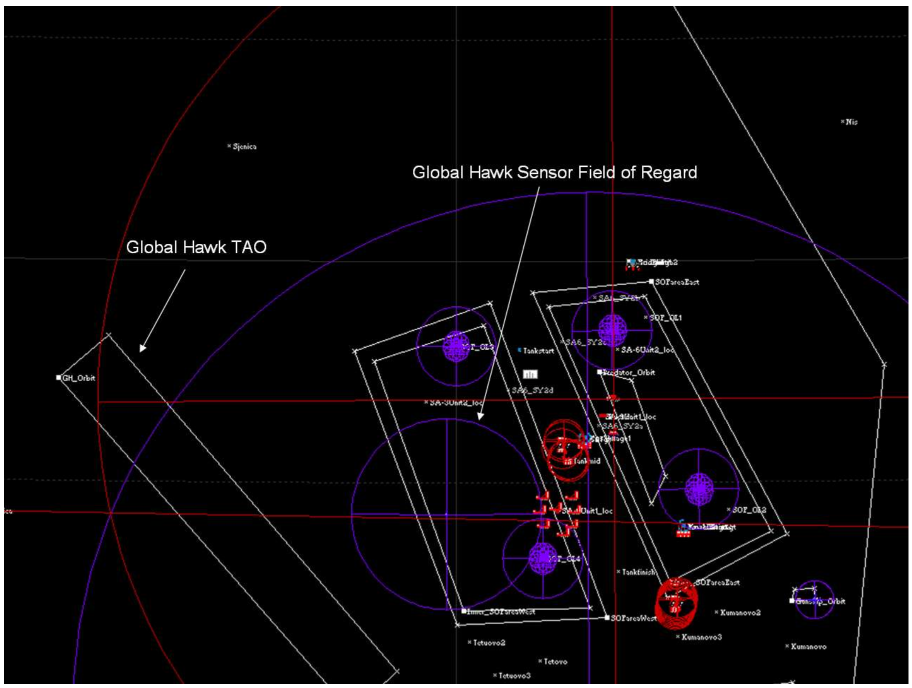

Figure 3.12 Global Hawk Diverting from TAO to Investigate a Target

SEAS provides a details window which allows the analyst to view agents status (dead or alive), current location, goal location, and number of targets. The debug window provides a view of any agents global and local variable values, attribute values, and local target list at every time step. The debug window proved invaluable in the validation of agent orders. Used in concert these methods and tools allowed a sound model of the TCT activities to be represented.

3.5 Analysis Plan

Now that a basic model of TCT is complete the aforemention MUA of the architecture can proceed. One hundred monte carlo runs of each the baseline model from SMC and a basic TCT model built as defined in this section are performed. The replications provide a distribution of the events of interest occurring over time. Standard MOEs will be collected and evaluated, and other analysis methods are employed to capture the effectiveness of TCT.

3.6 Summary

The overall goal of generating an executable model of a system represented in the DoDAF, has been accomplished. Complete data products would offer opportunity for a more automated, and rigorous transformation of data. However, sufficient data was available to capture the overall methods and effects of TCT.

Figure 3.13 gives a high level view of the data transfer from the architectures to SEAS. Data available from the OV-3 (or SV-6) and SV-2 produced the communication structure in SEAS. Aggregation of the communication channels restricts the ability to evaluate attacks on communications. To properly identify such effects as jamming of the UHF band, information that flows over these channels is necessary. This is not a flaw of the DoDAF, just a result of an incomplete architecture.

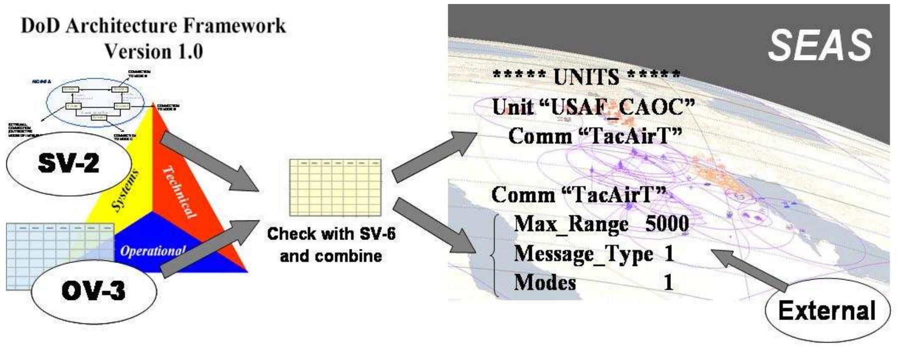

Figure 3.13 Architecture Data Products Producing Communications in SEAS Warfile

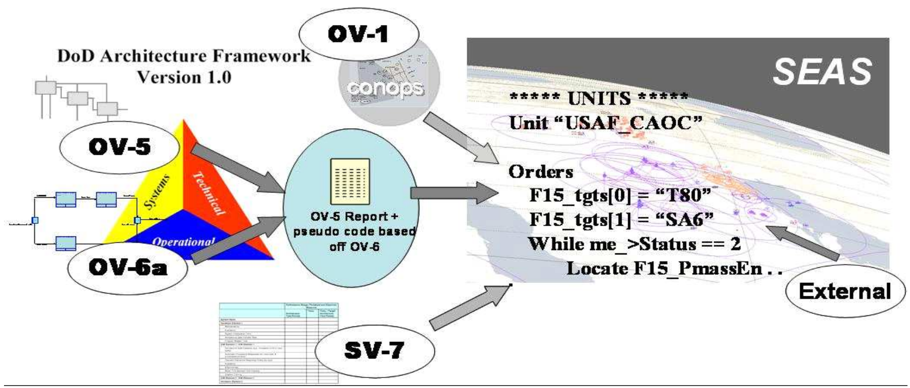

Figure 3.14 Architecture Data Products Producing Orders in SEAS Warfile

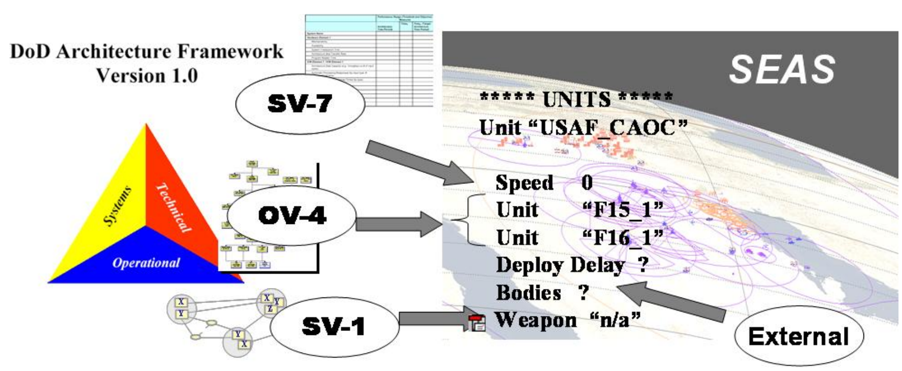

Figure 3.15 Architecture Data Products Filling General Attributes in SEAS Warfile

Figure 3.14 displays the products used to create orders in SEAS. Creating orders in SEAS to represent the TCT thread was restricted by no available OV-6 for the AOC architecture, and limitations in SEAS. Information for decisions to be made in TCT activities are generated externally or via activities in the AOC. Figure 3.15 yields the data products used to create general attributes for the AOC agent in SEAS. Filling the general attributes into SEAS is data TBD in DoDAF. While the TCT thread is not completely mapped into our SEAS model, we demonstrate and assess the retasking of ISR assets in Chapter 4.

4. Analysis

4.1 Overview

This chapter describes the results of the evaluation of the Baseline model, and the comparison of the Baseline and TCT models. The nature of the scenario and comparing the two systems are main drivers in the measures discussed here. First, the Kosovo-type situation calls for direct measures of military utility such as a reduction or cease in civilian casualties while keeping the number of blue causalities to a minimum. Also, a comparison of systems typically involves collecting measures like performance trade-offs and cost. While this information may be obtained via traditional MOE output of SEAS (number of bodies, vehicles, weapons, and dollars lost), trends in agent kills over time have also been captured in this study. This allows for possible identification of emergent enemy behaviors, successful periods of TTPs, a proper length of time to run the simulation, and possible classification of phases of the war.

4.2 Baseline Scenario Analysis

As mentioned in the discussion of SEAS, each agent is given a set of orders. These orders coupled with underlying default agent behaviors, and interaction with other agents guide the agent through the simulation. While it would be exhaustive to define each rule set and its implications on the scenario some important high level operation information is given here. Each Forces' Concept of Operations (ConOps) for the baseline model have remained unchanged from that received from SMC/TD. The USAFE ConOps do not retask ISR assets. This means each ISR asset is given a *TAO* to fly for the duration of the scenario. Also, the ranking of target priorities for the F15s remain constant over time. These priorities are lumped into a primary group including radar vans, and a secondary target group including all other targets.

For more detail one can refer to the discussions in Chapters 2 and 3 on SEAS, or the entire warfile can be found at AFIT's Center for Operational Analysis.

A set of exploratory simulation runs were completed to determine an appropriate length of time to run the simulation. These runs suggested no significant activity occurred after 6000 minutes of simulation time, and no event based criteria to stop the simulation was uncovered (e.g. all Serbian forces are killed or withdrew). Therefore, the time to end the simulation was set at one hundred hours (6000 minutes). To provide enough data points to properly compare this baseline with the TCT one hundred replications of the simulation were run. This also reduces the variability while evaluating events within the Baseline mode.

The total count of agent kills over the one hundred replications is used in the following plots to describe and compare the models. The total count is used to illustrate the behavior of the war over time as any one replication has a very small number of kills. Figure 4.1 displays the total number of kills of each class of platform agent over the scenario run time. This overall picture of kills vs. time was particularly useful in identifying two phases of the war. Phase I, origin of the war to 48 hours, is considered a SEAD phase. Figure 4.2 displays the high density of red radar kills as compared to other activities in the scenario. Phase II is considered as intervention of killing on the ground (see Figure 4.3). This phase is highlighted by a large distribution of Kosovar kills as opposed to other activities occurring during this time.

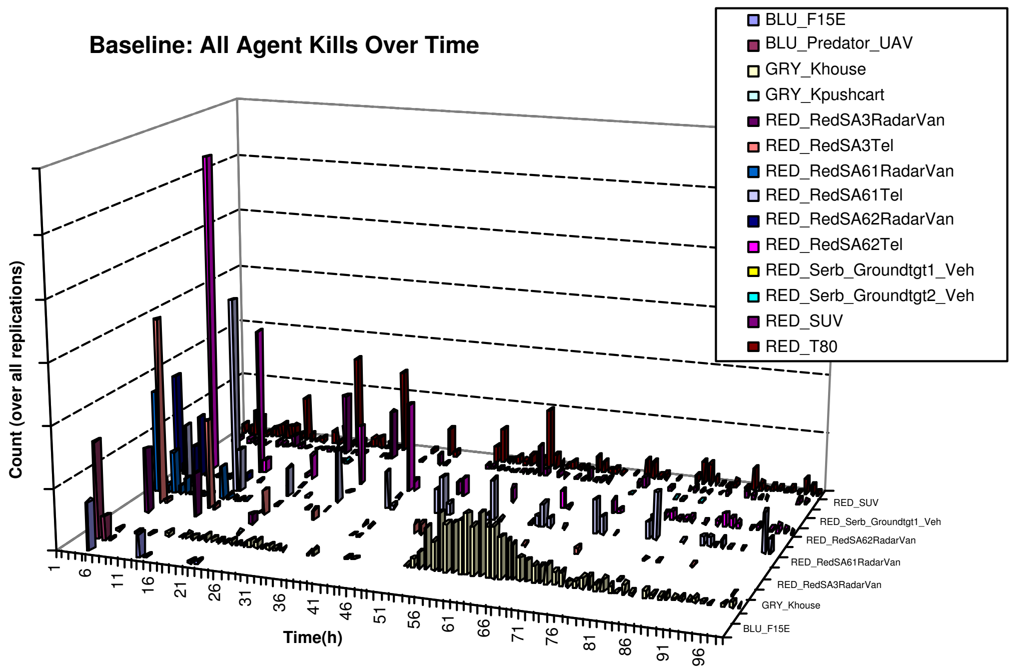

Figure 4.1 SMC Baseline

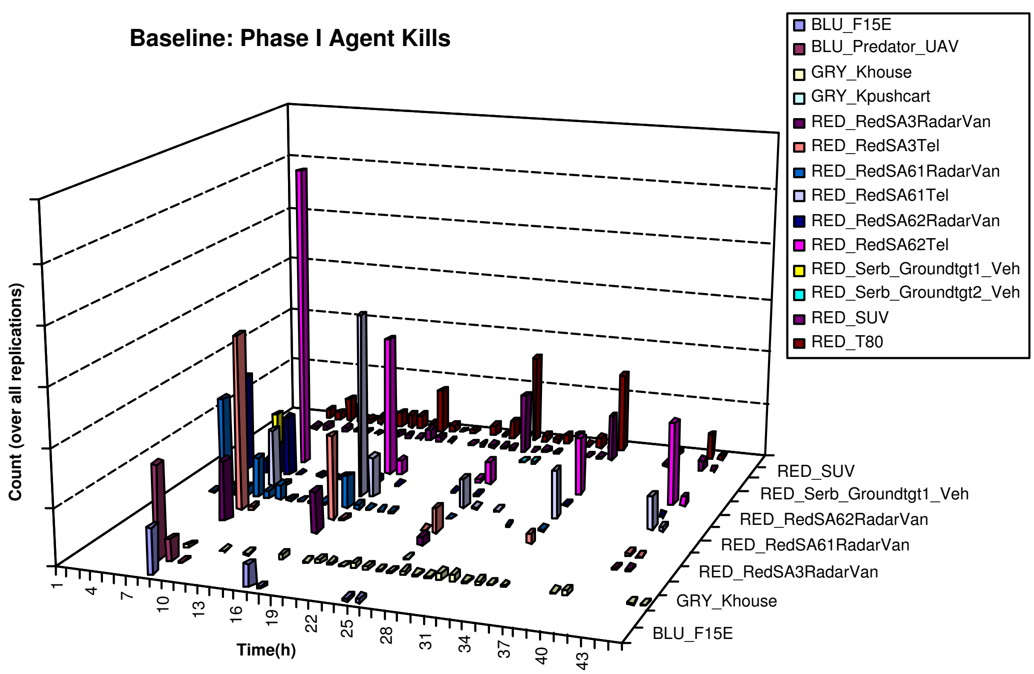

Figure 4.2 SMC Baseline: Phase I

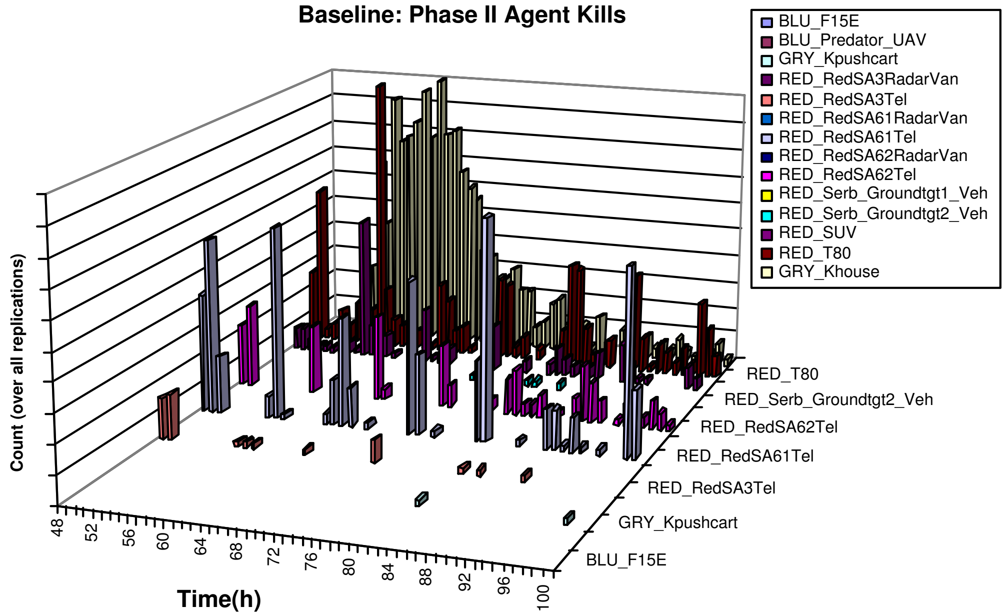

Figure 4.3 SMC Baseline: Phase II

To highlight the key factors in the outcome of the scenario, Figures 4.4 and 4.5 display time and quantity measures of Kosovar agent kills. Each Kosovar house and pushcarts platform agent has six and two associated bodies respectively. The number of Kosovars killed is also relative to how the scenario was modeled. Since SEAS has been coded with a vertical slice of the forces present, the number of kills experienced would be some multiple of what we see here. In the case of the F15 platform agent, Figure 4.1 shows early kills while the SAM threat is not yet suppressed. A further looks shows most kills are from the SA3, with an average of 1 F15 killed per replication. By identifying that there is no killing of F15 agents after Phase I, it may be plausible to change the priority of the F15 to kill Tanks and SUVs. The TCT model with the concept of a DTWL priorities of targets may change over time. Furthermore, the retasking of ISR assets may also change over time to aid in the detection of the tanks.

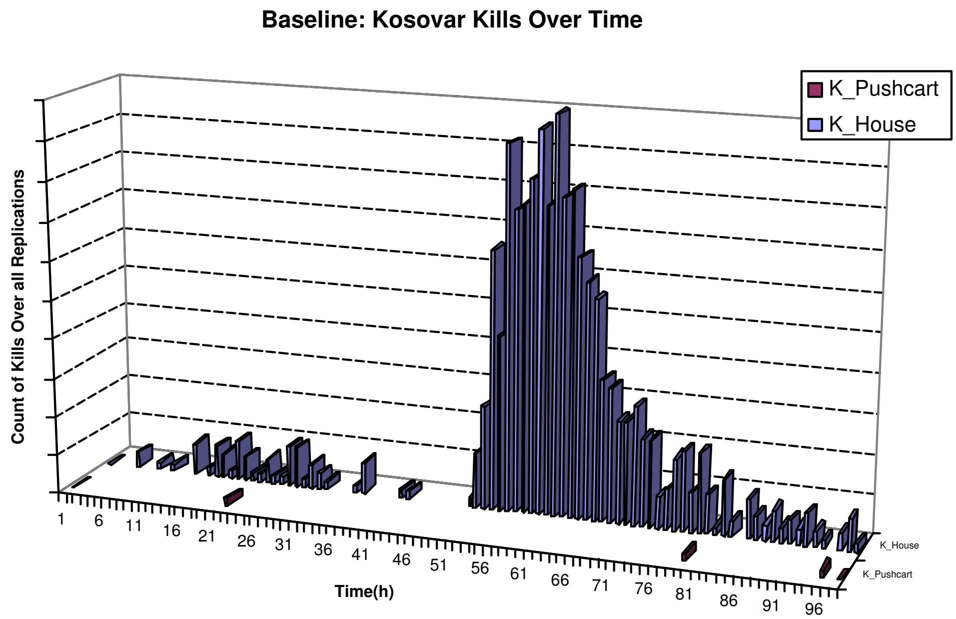

Figure 4.4 SMC Baseline: Kosovar Kills

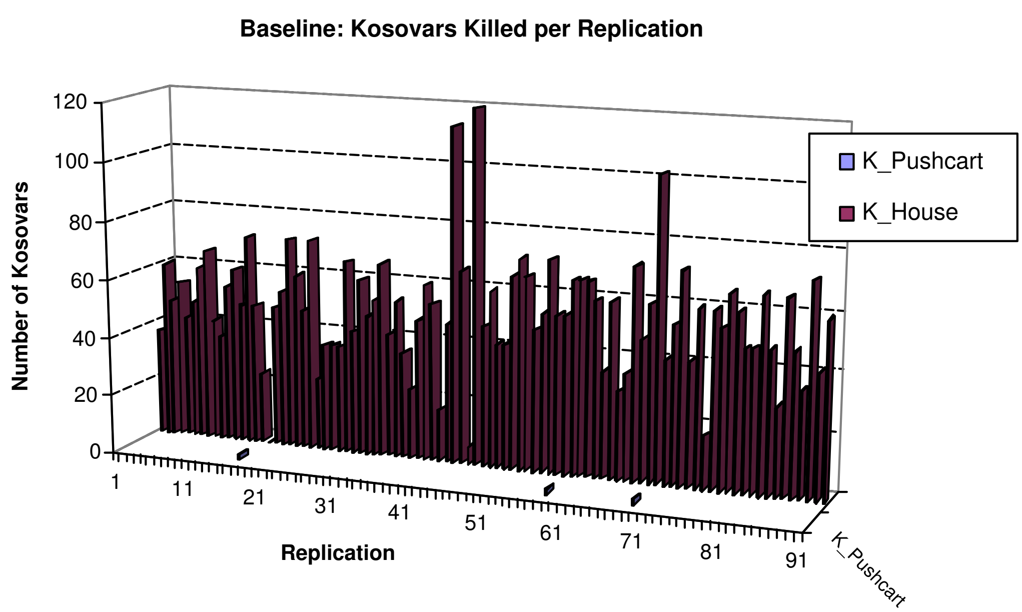

Figure 4.5 Number of Kosovars Killed per Replication

4.3 Model Comparison

The TCT model is run for one hundred replications beginning with the same random number seed as the Baseline model. All changes and evaluations made in chapter 3 have been implemented. Also, the F15 and ISR agents (global hawk and predator) reprioritize targets to kill or detect 48 hours after the simulation has started. Also, the F15s DTWL prioritization will then be changed to:

$$
\begin{aligned}
\textit{F15\_DTWL[0]} &= \text{“}\textit{T80}\text{”} \\
\textit{F15\_DTWL[1]} &= \text{“}\textit{Serb\_Groundtgt3\_Veh}\text{”} \\
\textit{F15\_DTWL[2]} &= \text{“}\textit{Serb\_Groundtgt2\_Veh}\text{”} \\
\textit{F15\_DTWL[3]} &= \text{“}\textit{Serb\_Groundtgt1\_Veh}\text{”} \\
\textit{F15\_DTWL[4]} &= \text{“}\textit{RedSA61RadarVan}\text{”} \\
\textit{F15\_DTWL[5]} &= \text{“}\textit{RedSA62RadarVan}\text{”} \\
\textit{F15\_DTWL[6]} &= \text{“}\textit{RedSA3RadarVan}\text{”} \\
\textit{F15\_DTWL[7]} &= \text{“}\textit{RedSA61Tel}\text{”} \\
\textit{F15\_DTWL[8]} &= \text{“}\textit{RedSA62Tel}\text{”} \\
\textit{F15\_DTWL[9]} &= \text{“}\textit{RedSA3Tel}\text{”}
\end{aligned}
$$

The global hawk and predator are retasked from their established orbits in two phases. In Phase I they fly *Investigate_TAO*s supporting the identification of SA radars, and in Phase II they support identification of T80 tanks and Serbian ground targets. This is to more appropriately capture the types of output from activities of DTWL prioritization and ISR Retasking (see Figure 3.11).

Figure 4.6 compares the total count of Kosovar agents kills in the Baseline and TCT models. The TCT model show a slight reduction in Kosovar kills with no significant shift in the distribution of kills over time. A comparison of the agents responsible for killing the Kosovars can be seen in Figure 4.7. The TCT model shows an overall greater number of T80 tank agent kills, but again no large shift in the distribution of kills over time.

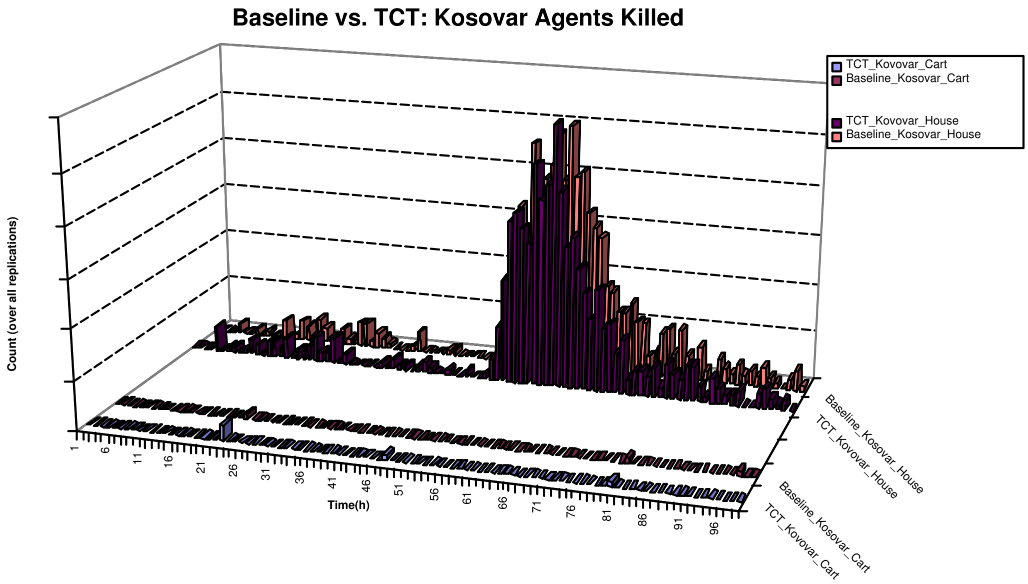

Figure 4.6 TCT vs. Baseline: Kosovar Agent Kill Comparison

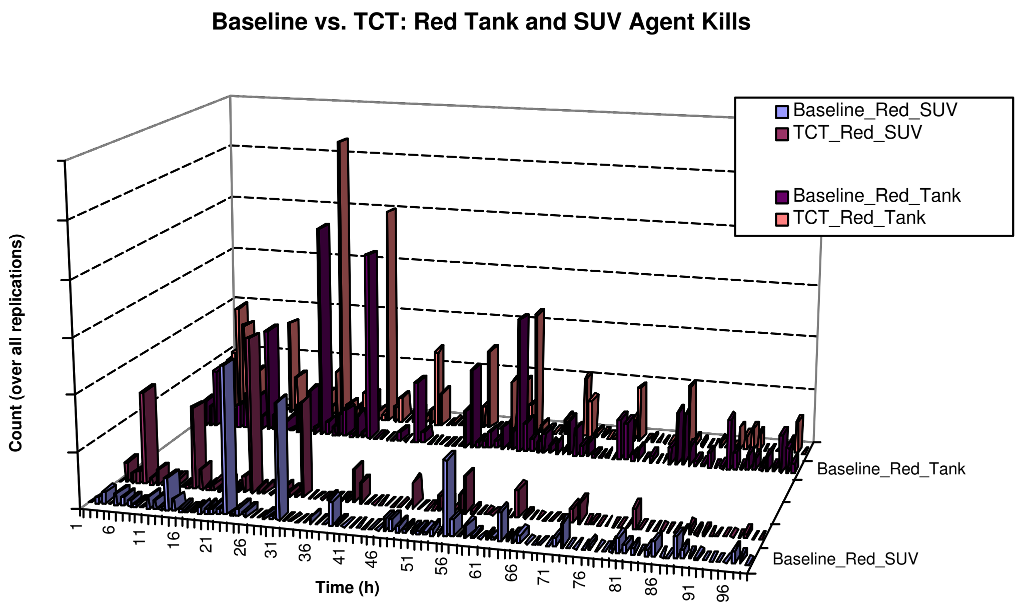

Figure 4.7 TCT vs. Baseline: Red Agent Kill Comparison

Table 4.1 Average and Differences of F15 Kills for Five Replications of the Two Models

j	X_{1j}	X_{2j}	Z_j
1	.35	.8	-.45
2	.8	.4	.4
3	.85	.4	.45
4	.3	.55	-.25
5	.95	.15	.8

To determine if the number of kills over all agent types has been significantly reduced or increased from the baseline scenario by employing the TCT activities a paired-t confidence interval approach is used. The approach allows us to compare the expected responses by forming confidence intervals for the difference in the two expectations. Table 4.1 yields the necessary statistics collected in SEAS to calculate the *paired-t* confidence interval for the number of F15 kills. The one hundred runs for each model are grouped into $j = 5$ replications of twenty runs. The average number of F15 kills per replication are represented by $X_{i,j}$ where i = the model (1 = Baseline, 2 = TCT) and j = replication. Each replication's average is then subtracted from one another, $X_{1,j}$ - $X_{2,j}$, to yield Z_j where j is as previously defined.

We then assume the $Z_j's$ to be independent and identically distributed (IID) random variables. Next, an interval about the expected values of the differences is calculated. The the expected value of the $Z_j's$, is

$$\overline{Z}(n) = \frac{\sum_{j=1}^{n} Z_j}{n}, \tag{4.0}$$

and the approximate $100(1-\alpha)$ percent confidence interval is defined by

$$\overline{Z}(n) \pm t_{n-1,1-\frac{\alpha}{2}} \sqrt{\frac{\widehat{Var}[\overline{Z}(n)]}{n}}. \tag{4.1}$$

Table 4.2 Paired t-test for All Agents Killed

Agent Type	$\overline{Z}(n)$	$CI\_LowerBound$	$CI\_UpperBound$	$Reject\ \emptyset$	$Model\ in\ Favor$
F15	.95	.4529	1.447	Yes	TCT
Predator	.1	.0739	.1261	Yes	TCT
Kosovar House	4.05	3.242	4.857	Yes	TCT
Kosovar Cart	-.1	-.1723	-.2770	Yes	Baseline
SA3 Radar Van	0	0	0	No	N/A
SA3 Tel	.1	.0272	.1728	Yes	Baseline
SA61 Radar Van	0	0	0	No	N/A
SA61 Tel	-.3	-.6371	.0371	No	N/A
SA62 Radar Van	0	0	0	No	N/A
SA61 Tel	1.5	1.115	1.884	Yes	Baseline
Ground Tgt 1	-2.35	-2.503	-2.197	Yes	TCT
Ground Tgt 2	-.85	-1.003	-.697	Yes	TCT
Serbian SUV	-1.1	-1.977	-0.223	Yes	TCT
Serbian Tank	-3.3	-5.651	-.9488	Yes	TCT

In the case when $\alpha = .1$ and $\overline{Z}(n) = .95$ the result is a lower CI bound of .453 and upper CI bound of 1.447. This leads us to reject the null hypothesis that there is no difference between the models in the number of F15 kills. We can further state that the TCT model leads to a statistically significant lower average number of F15 kills at 90% level of confidence.

This same procedure was applied to all platform agents killed in the scenario and results are shown in Table 4.2. The table gives the expected value of the difference, $\overline{Z}(n)$,along with the CI bounds. A Yes/No decision to reject the null hypothesis based on whether the CI contains zero is made. The difference in kills, if applicable, is then credited to which scenario it favors. For instance, we always would like to see more SA6 tels killed. The SA61 Tel agents $\overline{Z}(n) = 1.5$ which tells us that we expect to see 1.5 less SA6s killed in the TCT model thus giving this measure in favor to the Baseline.

While Table 4.2 shows most kills are in favor of the TCT model the Baseline model does show an advantage in the number of SA Tel and Kosovar Cart kills. The total number of Kosovar Cart kills over all replications is an extremely low count (three for the Baseline model), so little analysis or interest is paid to this comparison. In the matter of the SA Tel kills the answer is a matter of a trade-off in attacking the more lethal SA3 radar with the highest priority as opposed to attacking all SA radars with the highest priority (most tel kills are a result of the F15 air-to-ground radar picking up the tel while attacking a near by radar van). When considering these differences in agent kills we must also take into account the operational significance of the difference. The differential of .1 SA3 Tels will most likely have no impact on the scenario, and 1.5 additional SA6 Tels are of limited use if the SA6 radar van is no longer operational. Also when considering operational impact in the TCT scenario each Kosovar House agent includes six civilians. The reduction, on average, of 4.05 Kosovar houses per replication is reducing the number of dead Kosovar civilians by 24.

The TCT activities utility and impact on the scenario is not solely reflected in outcomes like number of kills. Other measures such as the effectiveness of sorties flown needs also to be addressed. Figures 4.8 and 4.9 displays a comparison, over time, of the number and effectiveness of sorties respectively. From these figures we see an increase in sortie effectiveness that is most noticeable during Phase II. Also, the Baseline model produced 6661 total F15 sorties as opposed to 5111 total sorties produced in the TCT model. The greatest reduction in the number flown is also seen in Phase II.

Insight from the sortie comparison data (less and more effective sorties) is incomplete without measures on the number of kills. This is due in part since SEAS calculations on sortie effectiveness are based on the number of bombs dropped, and not the number of targets hit. From Table 4.2 we see that most kills are reduced or in the case of killing tanks increased in favor of the TCT model. The practical significance of these changes is left to the decision-maker. However, the same or even better overall mission outcomes are obtained with less sorties flown.

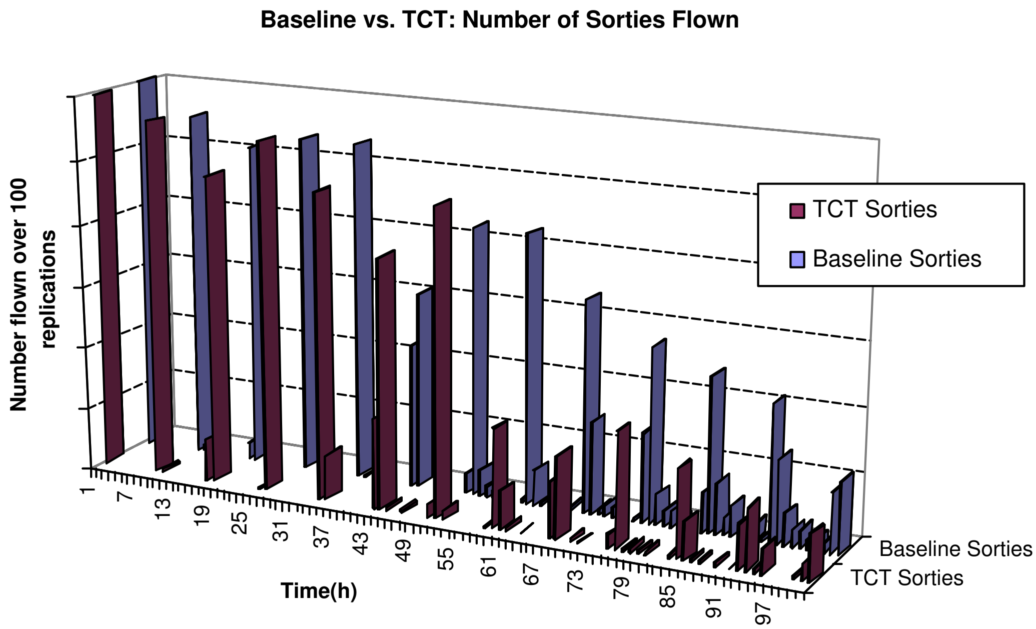

Figure 4.8 TCT vs. Baseline: Total Sorties Flown

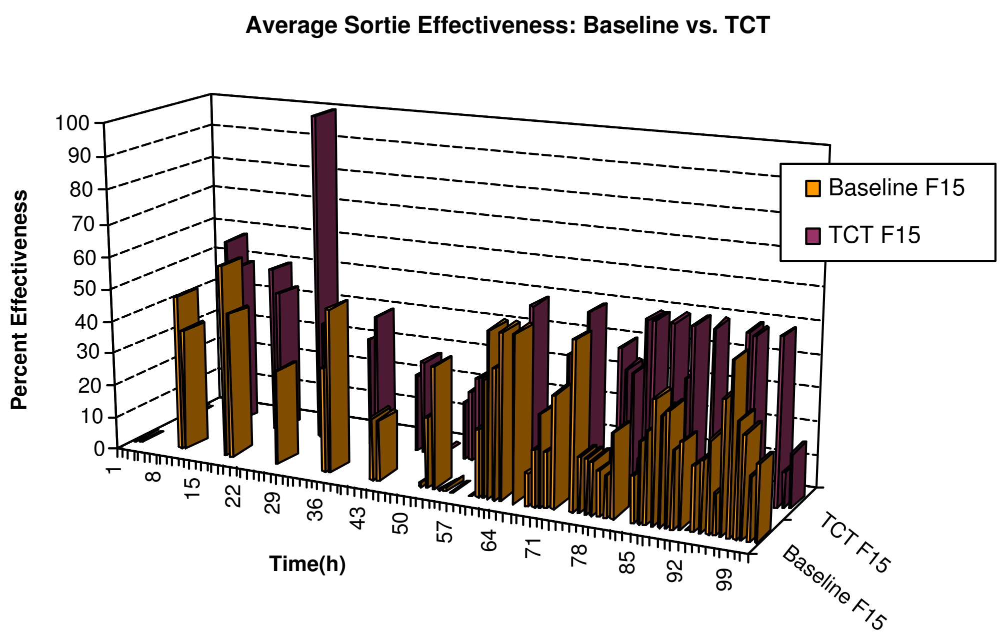

Figure 4.9 TCT vs. Baseline: Sortie Effectiveness

4.4 Summary

The results from this chapter demonstrated the modeled TCT activities had significant impact on the number of agent kills, and sorties flown in the Kosovo scenario. Initial analysis of behavior over time identified two phases of the war. From this the F15 priorities and ISR retasking in the TCT model were programmed to be updated after the first phase. These updates coupled with the modeled TCT activities favorable statistically significant changes in agent kills are realized. Further analysis illuminated the fact fewer and more effective sorties accomplish these more favorable outcomes in the TCT model.

5. Conclusions

5.1 Overview

This research utilized agent based simulation to create an executable model of a C4ISR weapon system represented within the DoDAF. This is done by the identification of possible methods of consistent integration of Operation and System View products into SEAS. The TCT model of the AOC and its operations were built from these views found in the architecture. Consistent integration of these products is integral to a valid comparison of the relative utility of alternative architectures and TTPs were the TCT model would effectively serve as an "approved baseline". This chapter presents a summary of the utility analysis of the AOC as represented by the TCT key thread architecture when placed within the SEAS Kosovo scenario. Differences realized between the TCT and Baseline models provide justification for utilizing architectures, when available, as viable source documentation for M & S. Limitations due to information not yet completed in the architecture are addressed. Also, recommendations for SEAS to be more receptive to architectural view products are presented. Finally, suggestions for future research are made.

5.2 Summary of MUA

Comparisons of the two models, Baseline and TCT, are measured by data extracted from one hundred replications of each scenario. Traditional MOEs are useful in describing the overall combat outcome (i.e. number of bodies lost over the entire war). This research also extracted measures over time uncovering trends of behavior to be exploited by the AOC activities described in TCT thread.

The number of agent kills over time illuminated two phases of the war classified as SEAD and Intervention of Killing. After identifying these phases, focus on shifting the TCT thread architecture led us to F15 priorities and ISR retasking at the 48 hour point. In the absence of a human (or more realistic modeling of the AOC) this shift was intended to capture some decisions made in analyzing targeting opportunities and readjusting theater ISR support. These changes being made, along with sufficiently capturing other TCT activities, the TCT model is executed and then compared.

Results show the Blue force in Phase I is effective for both the baseline and the TCT mission. The casualties of Blue force agents (F15 and Predator) are minimal and SA kills high, but some subtle differences are evident. The SA3 rather than the SA6 proved to be more lethal to the F15. The TCT model focuses its early missions on the SA3 radars influencing the reduction of overall F15 kills. The number of F15 kills are shown to be statistically significantly different via the t-test performed (see table 4.2).

Phase II analysis displayed no divergent behavior in the time of kills between the models (i.e. the TCT model did not influence the Serbian forces to interact with Kosovar civilian later in the war). However, the TCT model displayed overall reductions in the number of Kosovars, Serbian ground targets, Serbian tanks, and Serbian SUVs killed. Also, the overall number of sorties flown in the TCT model are significantly less, with sorties flown in Phase II being significantly more effective. While effectiveness is calculated by number of weapons released per sortie, it can still be said that the same (or better) overall mission is accomplished with fewer sorties by the TCT model.

Utilizing the ability to focus attacks, retask ISR assets, and support missions with continuous ISR support the TCT model provided reductions in friendly kills along with higher mission effectiveness. More important is illuminating the impact of modeling the activities provided in the architecture. If the architecture is represented without or misrepresents these activities significant impacts on the outcome of the war would not have been realized. This give credence to utilizing architectures as source documentation for M & S.

5.3 *Limitations with Architecture*

The original AOC architecture provided an OV-5 product, but did not provide an OV-6 product. The OV-5 provided enough information to be aware of what information flows in and out of an activity, but no rules or sequencing of rules for these activities are available without an OV-6 product. The solution was the TCT "key-thread", a detailed subset of the AOCs activities sharing the same information in its creation. The TCT architecture provided an OV-6, operational rules model, allowing sequence of events and rules to be modeled. However, this transfer of data was not without its problems.

The IDEF3 format of the OV-6 defines decision points and junctions quite well, but generally lacked information on entry/exit conditions for processes and thresholds for decisions. For example Figure 3.10 begins with a decision to be made if significant movement in the target is detected. No threshold on this movement is defined in order to make a decision. This is due to the decision being qualitative in nature, and is left to the analyst to set proper criteria based on the scenario modeled. With this flexibility lies the potential for inconsistent modeling of the activity.

Each activity is completed by a collection of systems. In order to model overall system performance we require performance parameters for each system involved in the activities. We are interested in some time delay for each individual system to complete its job and to accumulate these values to yield some overall system performance. SEAS is receptive to this as we can add delays in the orders. Any time an agent cycles through an IF loop or calls a function we can add the delay. Again the current version of the AOC architecture does not provide an SV-7 product which would contain this desired information.

Also, from the architecture provided we are not able to completely represent the communications network. The OV-3 product provided us with the information needed to be exchanged between nodes (nodes being represented by agents in SEAS).

The OV-3 also defines through which of the nodes' activities the information exits and enters. The SV-2 informs us of the systems to be used in these exchanges. However, we are not given sufficient information to discern which systems are to transmit which types of information. In effect we can not effectively identify bottlenecks or jamming effectiveness. For example, we can not correctly identify the impact of only UHF jamming if all information between two nodes is degraded due to the aggregation of the communication channels between nodes.

5.4 Recommendations for SEAS Improvement

In the effort to make SEAS more receptive and the information more consistent from architecture products, some suggestions are made. In this research building and verifying the communications network in SEAS became difficult. It is clear from the SEAS help where and how to input the communication equipment in the warfile. Establishing a link between two agents requires both agents to own the same communication equipment. However, in a complicated network it becomes difficult to identify when all agents have been linked correctly. To aid us in this study the network graphs seen in Figures 3.4 and 3.5 were used. After the graphs were established from the architecture data, we began the manual process of matching the links, modes, and message types to the warfile. We suggest an GUI interface to SEAS where agents can be placed on a pallet. Once agents are placed they can be linked by coded communication arrows thus creating a picture much like the one created in the figures. The information can be then be saved to the warfile automatically. This would reduce the analysts time to represent the network in SEAS, and provide a first step in verifying the transfer.

Next, in the case of modeling the TCT activities we identified a need for more flexible assignments of probabilities in SEAS. In Activity 7, Validation of Target, (in the case of a first strike), is left to the PI_d. If the detecting sensor provides a PI_d high enough the F15 squadron may release an F15 to fly to the target. However, for the weapon to engage the target a PI_d_Commit threshold must be met. The SEAS help and user briefs do not define how these two probabilities are related. In simple experiments completed in this research it is concluded that a PI_d from a single sensor must be greater than the PI_d_Commit for the weapon. Additional sensor PI_ds are not accumulated to create a higher PI_d. There may be cases where multiple sensor information, or a long dwell time on the target warrant a better perception of the target thus a higher PI_d. Compensation for these cases along with some clarification of these probabilities in SEAS is recommended.

Finally, it has been mentioned that the vertical slice aggregation technique allow us to capture C4ISR effects by the ability to track the sensor to shooter chain of events. The standard SEAS killer victim scoreboard output file makes it quite easy to define who an agent is killed by, but not so clear as to what sensor was responsible for the sighting resulting in the kill. All sensor detections are available in the communications output file, but deciphering the chain of events from sensor detection to kill is difficult.

5.5 Future Research

The AOC agent in this case study was modeled in a very simplistic manner. The fidelity at which the AOC can be modeled is dependent upon the experience of the SEAS analyst. In 2001 the Rand Corp. sought to improve the C4ISR capabilities in SEAS [24]. In this study the AOC is detailed and well represented, but created from out of date information. Also, the study utilized a previous version of SEAS. However, the level of effort and detail in this study should be the focus of future modeling of the AOC.

Due to the limitations of the provided architecture we were unable to assess the role of the SV-7 information. It was assumed this data would allow us to build the communication network to a level of detail needed to asses jamming and bottleneck studies. By repeating this case study we could validate the role of this information. An objective of this study was to minimize analyst intervention and time in the transfer of data from an architecture to a combat model. Follow on research can focus on a more automated process for the information transfer of data products that proved to have a valid place in the warfile. Both the System Engineer and SEAS Analyst recommend future undertakings include a computer programmer as the automation would require manipulating report generation scripts into VBA or C+ code. Finally, analysis completed in SEAS may provide recommended changes to the modeled C4ISR system. In the automation of data transfer we recommend incorporating the flexibility of a transition from SEAS, or another combat model in general back to architectures.

Appendix A. List of Acronyms

ABCM	Agent Based Combat Model
AFDD	Air Force Doctrine Document
AFSAT	Air Force Standard Analysis Toolkit
AOC	Aerospace Operations Center
ATO	Air Tasking Order
AV	All View
BDA	Battle Damage Assessment
C4ISR	Command Control
CAOC	Combined Aerospace Operations Center
CAS	Complex Adaptive System
ConOps	Concept of Operations
DMSO	Defense Modeling and Simulation Office
DoD	Department of Defense
DODAF	Department of Defense Architectural Framework
EBO	Effects Based Operations
FEBA	Forward Edge of the Battle Area
ICOM	inputs, controls, outputs, and mechanisms
IDEFØ	method designed to model decisions, actions, and
IDEF3	Process Flow and Object State Description Capture Method
IID	Independent and Identically Distributed
IS	Information Superiority
JIPTL	Joint Integrated Prioritized Target List
MOE	Measure of Effectiveness
MUA	Military Utility Analysis
OODA	Observe Orient Decide Act
OV	Operational View

Pd	Probability of Detection
Pk	Probability of Kill
ROC	Receiver Operating Characteristic
ROE	Rules of Engagement
SA	Surface to Air
SAM	Surface-to-Air Missile
SEAD	Suppression of Enemy Air Defense
SEAS	System Evaluation Analysis Simulation
SMC/TD	Space and Missile Center Transformations Directorate
SOF	Special Operations Force
SV	System View
TBMCS	Theater Battle Management Core System
TCT	Time Critical Target
TPL	Tactical Programming Language
TPW	Target Planning Worksheet
TTP	Tactics Techniques and Procedures
TV	Technical View
USAFE	United States Air Forces Europe

Bibliography

1. "Air Force Basic Doctrine; Air Force Doctrine Document 1." Air University, September 1997.
2. Admiral Arthur K. Cebrowski U. N. and Garstka J. J., "Network-Centric Warfare: Its Origin and Future." web, 1998.
3. Bullock R., McInyre G., and Hill R. "Using Agent-Based Modeling To Capture Airpower Strategic Effects." *Proceedings of 2000 Winther Simulation Conference*. 2000.
4. Caldwell B. and et. al , "Aggregated Combat Models." Navy Postgraduate School Operations Research Department, Febraury 2000.
5. CNA , "EINSTein/ISAAC White Paper." web, 2004.
6. Davis P. K. *Aggregation, Disagregation, and the 3:1 Rule in Ground Combat* (First Edition). Santa Monica CA: RAND Corp., 1995.
7. Davis P. K. *Analytic Architecture for Capabilities-Based Planning, Mission-System Analysis, and Transformation* (First Edition). Santa Monica CA: RAND Corp., 2000.
8. DoD , "C4ISR Architecture Framework Version 2.0.." 4.6.1, December 1997.
9. DoD , "DoD 5000.9-D.Department of Defense Modeling and Simulation (M & S) Master Plan,." Undersecretary of Defense for Acquisition and Technology, January 1994.
10. DoD , "DoD 5000.9-M.Glossary of M & S Terms." Undersecretary of Defense for Acquisition and Technology, January 1994.
11. DoD , "Department of Defense Directive 5000.2, Operation of the Defense Acquisition System." Undersecretary of Defense for Acquisition and Technology, May 12, 2003.
12. Gonzales D., Moore L., Pernin C., Matonick D., and Dreyer P., "Assessing the Value of Information Superiority for Ground Forces - Proof of Concept." RAND Corp., 2001.
13. Handley H., Zaidi Z., and Levis A., "The Use of Simulation Models in Model Driven Experimentation, System Architectures Laboratory, C3I Center." System Architectures Laboratory, C3I Center, George Mason University, 1999.
14. Hartman J. K., "Lecture Notes In Aggregated Combat Models." classroom distributions, 1985.

15. Hillestad R. J. and Juncosa M. L. *Cutting Some Trees to See the Forest: On Aggregation and Disaggregation in Combat Models* (First Edition). Santa Monica CA: Rand Corp., 1993.

16. Hillestad R. J., Owen J., and Blumenthal D. *Experiments in Variable Resolution Combat Modeling* (First Edition). John Wiley & Sons, Inc., 1995.

17. Horrigan , "Configuration and the Effectiveness of Air Defense Systems in Simplified, Idealized Combat SituationsA Preliminary Examination." Horrigan Analytics, Chicago IL, June 1995, 1995.

18. Ilachinski A., "Irreducible Semi-Autonomous Adaptive Combat (ISAAC):An Artificial-Life Approach to Land Combat, White-Paper." Center for Naval Analysis, Alexandria VA, 1997.

19. Ilachinski A. *Cellular Automata A Discrete Universe* (First Edition). Singapore: World Scientific Publishing, September 2001.

20. Jensen K., "Coloured Perti Nets." Computer Science Department University of Aarhus Slides, 1996.

21. Jervis R. "Complex Systems: The Role of Interactions." *Proceedings of the Complexity, Global Politics and National Security Conference*. http://www.ndu.edu/inss/books/booksics November 13-14 1996.

22. Levis A. and Wagenhals L., "C4ISR Architectures: I. Developing a Process for C4ISR Architecture Design." System Architectures Laboratory, C3I Center, George Mason University, July 200.

23. Ling M. F., "Agent-based Modelling of Command-and-Control Effectiveness." web, 2001.

24. Moore L., Gonzales D., Matonick D., Pernin C., and Uy R., "Improving the C4ISR Analysis Capability of the System Effectiveness Analysis Simulation (SEAS)." Unpublished RAND Study, 2001.

25. Panel D. I. A., "IEEE STD 610.12." DoD, 1995.

26. Pawlowski T., Barr P., and Ring S., "Executable Architecture Methodology for Analysis." The MITRE Corporation unpublished slides, September 2003.

27. RAND , "Rand Brief to AFIT about SEAS." unpublished slides, 2002.

28. Sanchez S. M. and Lucas T. W. "Exploring the World of Agent-Based Simulations: Simple Models, Complex Analysis." *Proceedings of the 2002 Winter Simulation Conference*. 116–126. 2002.

29. Tighe T. R. *Strategic Effects of Airpower and Complex Adaptive Agents: An Initial Investigation* (First Edition). School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH: MS thesis, AFIT/GOA/ENS/99M-09, March 1999.

30. Walter Perry , Robert W. ButtonJ. B. T. S. J. M. *Measures of Effectiveness for the Information-Age Navy: The Effects of Network-Centric Operations on Combat Outcomes* (First Edition). RAND Corp., 2002.

31. Weber R., "Personal Corresondence." The Aerospace Corporation, El Segundo CA, June 2003 - March 2004.

32. Zinn A. *The Use of Intergrated Architectures to Support Agent Based Simulation: An Initial Investigation* (First Edition). School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB OH: MS thesis, AFIT/GSE/ENY/04-M01, March 2004.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
09-03-2004	**Master's Thesis**	Jun 2003 – Mar 2004

4. TITLE AND SUBTITLE
AGENT BASED SIMULATION SEAS EVALUATION OF DODAF ARCHITECTURE

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
DeStefano, Gregory, V., Captain, USAF

5d. PROJECT NUMBER
ENR2003-078

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Building 642
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GOR/ENS/04-05

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Space and Missile Center Transformation and Development Directorate
2430 East El Segundo Blvd
Los Angeles AFB CA 90245

10. SPONSOR/MONITOR'S ACRONYM(S)
SMC/TD; AFSAA

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
N/A

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES
N/A

14. ABSTRACT
With Department of Defense (DoD) weapon systems being deeply rooted in the command, control, communications, computers, intelligence, surveillance, and reconnaissance (C4ISR) structure, it is necessary for combat models to capture C4ISR effects in order to properly assess military worth. Unlike many DoD legacy combat models, the agent based model System Effectiveness and Analysis Simulation (SEAS) is identified as having C4ISR analysis capabilities. In lieu of requirements for all new DoD C4ISR weapon systems to be placed within a DoD Architectural Framework (DoDAF), investigation of means to export data from the Framework to the combat model SEAS began. Through operational, system, and technical views, the DoDAF provides a consistent format for new weapon systems to be compared and evaluated. Little research has been conducted to show how to create an executable model of an actual DoD weapon system described by the DoDAF. In collaboration with Systems Engineering masters student Captain Andrew Zinn, this research identified the Aerospace Operation Center (AOC) weapon system architecture, provided by the MITRE Corp., as suitable for translation into SEAS. The collaborative efforts lead to the identification and translation of architectural data products to represent the Time Critical Targeting (TCT) activities of the AOC. A comparison of the AOC weapon system employing these TCT activities with an AOC without TCT capabilities is accomplished within a Kosovo-like engagement (provided by Space and Missile Center Transformations Directorate). Results show statistically significant differences in measures of effectiveness (MOEs) chosen to compare the systems. The comparison also identified the importance of data products not available in this incomplete architecture and makes recommendations for SEAS to be more receptive to DoDAF data products.

15. SUBJECT TERMS
Combat Modeling, SEAS, System Architecture, DoDAF Architecture, Complex Adaptive Systems, Network, Kosovo, Aerospace Operations Center, Military Utility Analysis

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr. John O. Miller, AFIT/ENS
					19b. TELEPHONE NUMBER *(Include area code)*
U	**U**	**U**	**UU**	87	(937) 255-6565, ext 4326; e-mail: john.miller@afit.edu