NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

MODELING ANTI-AIR WARFARE WITH DISCRETE EVENT SIMULATION AND ANALYZING NAVAL CONVOY OPERATIONS

by

Ali E. Opcin

June 2016

Thesis Advisor: Arnold H. Buss
Co-Advisor: Thomas W. Lucas
Second Reader: Paul J. Sanchez

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704–0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704–0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** June 2016	**3. REPORT TYPE AND DATES COVERED** Master's thesis	
4. TITLE AND SUBTITLE MODELING ANTI-AIR WARFARE WITH DISCRETE EVENT SIMULATION AND ANALYZING NAVAL CONVOY OPERATIONS			**5. FUNDING NUMBERS**
6. AUTHOR(S) Ali E. Opcin			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943–5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release;distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT

Anti-air warfare (AAW) is a primary naval warfare area. Using AAW tactics and concepts of operations, this research explores the most critical success factors of convoy operations. In this study, a discrete event simulation (DES) was built by modeling ships, and their sensors and weapons, to simulate convoy operations under air threat. Where classified data was unavailable, assumptions were made and approximations were used in constructing the ships, weapons, and sensors. The model was used to simulate over 1.5 million naval battles varying 99 input variables using sophisticated and systematically created data combinations. To select the input settings over a specific range of input variables, a nearly orthogonal nearly balanced (NOB) Latin hypercube design was used. The effects of these input changes on the outputs were analyzed using partition trees and nominal logistic regression. The primary response variable was the survival of the High Value Unit (HVU) as a binary outcome. According to the analysis, in a convoy operation under air threat, the surface-to-air missile (SAM) specifications of the screen ships, the staying power of the HVU, and the anti-ship missile (ASM) specifications of the enemy ships had the most significant effect on the survival of the HVU.

14. SUBJECT TERMS Discrete Event Simulation, Modeling Anti-Air Warfare, Simkit, Component Based Approach, Layered Defense Systems, Formation Movements, Design of Experiments, Simulation Output Analysis			**15. NUMBER OF PAGES** 147
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540–01–280–5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release;distribution is unlimited

MODELING ANTI-AIR WARFARE WITH DISCRETE EVENT SIMULATION AND ANALYZING NAVAL CONVOY OPERATIONS

Ali E. Opcin
Lieutenant Junior Grade, Turkish Navy
B.S., Turkish Naval Academy, 2011

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH
and
MASTER OF SCIENCE IN MODELING, VIRTUAL ENVIRONMENTS, AND SIMULATION

from the

NAVAL POSTGRADUATE SCHOOL
June 2016

Approved by: Arnold H. Buss
Thesis Advisor

Thomas W. Lucas
Co-Advisor

Paul J. Sanchez
Second Reader

Christian Darken
Chair, Moves Academic Committee

Peter J. Denning
Chair, Department of Computer Science

Patricia A. Jacobs
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Anti-air warfare (AAW) is a primary naval warfare area. Using AAW tactics and concepts of operations, this research explores the most critical success factors of convoy operations. In this study, a discrete event simulation (DES) was built by modeling ships, and their sensors and weapons, to simulate convoy operations under air threat. Where classified data was unavailable, assumptions were made and approximations were used in constructing the ships, weapons, and sensors. The model was used to simulate over 1.5 million naval battles varying 99 input variables using sophisticated and systematically created data combinations. To select the input settings over a specific range of input variables, a nearly orthogonal nearly balanced (NOB) Latin hypercube design was used. The effects of these input changes on the outputs were analyzed using partition trees and nominal logistic regression. The primary response variable was the survival of the High Value Unit (HVU) as a binary outcome. According to the analysis, in a convoy operation under air threat, the surface-to-air missile (SAM) specifications of the screen ships, the staying power of the HVU, and the anti-ship missile (ASM) specifications of the enemy ships had the most significant effect on the survival of the HVU.

THIS PAGE INTENTIONALLY LEFT BLANK

THESIS DISCLAIMER

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

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AAW	anti-air warfare
ASM	anti-ship missile
ASW	anti-submarine warfare
AUC	Area Under Curve
BIC	Bayesian Information Criterion
CED	Close Enough Distance
CIWS	Close-In Weapon System
CSV	Comma Separated Values
DMM	Disposition Mission Model
DES	Discrete Event Simulation
DOE	Design of Experiments
HVU	High Value Unit
GUI	Graphical User Interface
LH	Latin Hypercube
MOE	Measure of Effectiveness
MOP	Measure of Performance
NOB	Nearly Orthogonal Nearly Balanced
NOLH	Nearly Orthogonal Latin Hypercube
RVFF	Resolution Five Fractional Factorial
ROC	Receiver Operating Characteristic
SAM	surface-to-air missile
SSAD	Ship Self Air Defense
SW	surface warfare
UML	Unified Modeling Language

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Convoy operations under various threats are among the most critical naval missions. Convoy operations are used to achieve many objectives, such as providing logistic support to a particular operational area or conducting an amphibious assault on enemy territory. While conducting convoy operations, the convoy and her escorts are exposed to many potential threats, including submarines, fighter airplanes, and/or surface ships, as well as the weapons that the opposing force uses. Therefore, the convoy's escorts need to implement anti-air, anti-submarine, and surface warfare tactics while conducting the operation. While all of these are important, this study focuses on anti-air tactics.

Simulation models are effective tools for analyzing naval operations. Modeling real life phenomena has many challenges, such as efficiently scaling the problem and systematically developing software. Verification and validation of the model and its inputs is also crucial. The model used in this study has not yet been validated. However, the simulation developed for this research, known as the AAW Analysis Model, provides a strong basis for analysis options, as it includes lots of design parameters. AAW is an acronym for anti-air warfare. Moreover, the AAW Analysis Model is scalable, modular, and flexible.

The AAW Analysis Model is built to analyze the effectiveness of a given screen disposition, screen ship properties, and High Value Unit (HVU) properties in convoy operations. It can also be used to identify the most effective factors in determining the success of convoy operations. The AAW Analysis Model is developed using the Simkit library in the Java programming language. It is also a unique model that incorporates the effects of screen disposition with a layered defense policy and surface warfare, including enemy ships and their engagement factors.

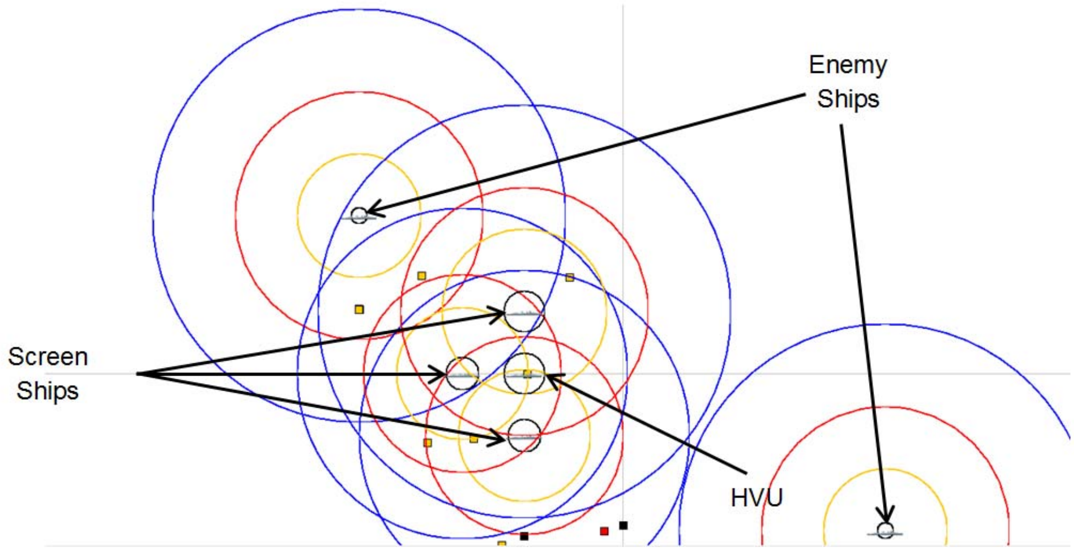


Figure 1. The AAW Analysis Model.

Because of the fact that the simulation includes many factors to analyze, an efficient design methodology was used to obtain systematically created input parameters. For the analysis portion of this research, the AAW Analysis Model was run for 1000 replications for each of 512 carefully-chosen design points in three different scenarios—resulting in over 1.5 million simulated naval battles.

The primary measure of effectiveness (MOE) extracted from the AAW Analysis Model is the survival of the HVU, a binary outcome that was recorded after each simulated battle. After the runs were made, partition trees and nominal logistic regression were used to build response surface metamodels to identify and quantify the most important factors on convoy operations under air threat.

The AAW Analysis Model includes hundreds of factors. Of them, 99 were chosen for exploration. There were 91 continuous and 8 discrete factors. Of these, 52 of them are controllable by the convoy and 47 of them are uncontrollable. Controllable factors include the HVU and screen ship properties. Uncontrollable factors relate to enemy ship properties.

The AAW Analysis Model was first analyzed across both uncontrollable and controllable factors to explore what the convoy can do and the enemy specifications that have the greatest effect on HVU survival. Subsequently, the outputs were analyzed over only the controllable factors in a robust analysis to identify the actions the convoy can take that are effective across a breadth of threat capabilities and tactics. The analysis determined that the anti-ship missile (ASM) specifications of enemy ships, the surface-to-air missile (SAM) specifications of the screen ships, and the HVU's staying power have the most significant effect on HVU survivability.

This study is just a first step in using the AAW Analysis Model to explore and enhance the safety of NATO ships in convoy operations where an air threat is possible. Further developments and modifications are needed for other types of operations or to explore all of the factors involved in this type of operational environment.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

The author is thankful to Prof. Arnold H. Buss, Prof. Thomas W. Lucas and Prof. Paul J. Sanchez for their contributions, guidance and patience during the work performing this research.

The author wants to express his gratitude for his family and his girlfriend, Ece Yildiz, for their great support.

Most importantly, the author is indebted to the Republic of Turkey and Turkish Navy for the opportunity to pursue a master's degree.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. PURPOSE

Anti-air warfare (AAW) is one of the primary aspects of naval warfare. Since the invention of mid and long-range missiles, missile usage and defense has been an important element in determining the outcome of naval wars. For instance, in the Falklands War between the United Kingdom and Argentina, the use of French-made Exocet missiles greatly impacted the war. In fact, the United Kingdom made an agreement with France to stop Argentina from acquiring more Exocets. Before that agreement, Exocet missiles previously obtained by the Argentinian Military were used to put the HMS Sheffield and Atlantic Conveyor out of action.

In naval warfare, especially for control of a sea area, there are three primary types of warfare that squadrons and task forces have to be aware of: anti-submarine warfare (ASW), AAW, and surface warfare (SW). ASW is outside of the scope of this study. SW is primarily based on anti-ship missiles (ASMs), which force warships to conduct AAW tactics and concepts to survive in an operational area. AAW and SW mostly depend on the sensors and close combat weapons that a warship has.

The quantity, availability, and capability of long and mid-range anti-ship missiles may pose a significant threat to NATO forces. Current anti-air missile tactics need to be developed further, and their future effectiveness is open to discussion (Townsend, 1999). Further scientific analysis of such tactics and systems is necessary using appropriate methods and tools. However, current combat analysis tools for naval anti-air warfare tactics have limited scope. They do not provide analysis equipment to comprehensively compare different tactics in AAW. Additionally, they mostly focus on single ship scenarios that are uncommon in AAW. Some naval combat models focus on the effects of formation movement. Nevertheless, their work does not fully evaluate the effect of enemy surface assets. Instead, they primarily focus on ASM raids.

B. BACKGROUND

AAW's purpose in naval operations may be defending a squadron or High Value Unit (HVU) from air threats. Air threats may be, but are not limited to, fighter airplanes or warships with anti-ship missiles (ASMs). AAW is also necessary to provide a protection for forces conducting naval convoy operations (O'Neil, 1981). These convoy operations include naval support for protection of commercial ships, amphibious forces, carrier task forces, and logistic carriers.

The first countermeasure against airstrikes consisted of mounting antiaircraft guns on ships. By the end of World War I, most of the important ships had a battery of one to four semiautomatic guns in high angle mountings, supplemented by machine guns. The machine guns were simple. The pilots fired them with simple computations without considering any factors that can increase accuracy, which is why they rarely hit their target (O'Neil, 1981). However, with the advance in technology, counter measures against airstrikes have become modernized. In contemporary naval warfare, modern combat ships and combat air patrols with fighter airplanes provide air defense for task forces with state-of-the-art weapons, such as modern guns and missiles. For instance, an aircraft carrier's combat air patrol is the most effective defense against enemy aircraft. Nevertheless, the screen ships that consist of frigates and destroyers can also provide a formidable defense against air threats.

Especially within a carrier group, air defense in naval tactics is often provided with layered defense tactics. At the center of these concentric layers, an aircraft carrier or other HVU is protected. A carrier strike group is shown in Figure 1. The outer layer usually consists of an airborne early warning and control system and combat air patrol that are composed of the fighter aircraft carried by the aircraft carrier. If an enemy force—which may be either an aircraft or a missile—gets into the air defense umbrella from this layer, then the next layers of defense are provided by aircraft based on the aircraft carrier that escorts the naval task force. Surface-to-air missiles (SAMs) are launched from surface platforms, such as the Standard Missile-1 (SM-1), with a range of up to 100 nm, and gun systems like 76 mm Oto Melara gun, with a range of up to 30 nm, provide point defense. As a last layer, a frigate or destroyer will usually be mounted with guns, including a Close-In Weapon System (CIWS) such as Sea-Zenith or Phalanx. A CIWS is a type of Gatling gun that can fire thousands of rounds in a minute. The calibers of those rounds are usually between 20 mm and 30 mm.

Figure 1. A Carrier Strike Group with Layers of Defense.
Source: (U.S.Navy, n.d.).

C. THESIS OBJECTIVES

The primary goal of this study is building a flexible, scalable, and expandable simulation model of naval AAW. This tool can also be used as a decision support tool to help decide which missiles or close combat weapons should be on particular types of combat ships. This model will also be a valuable tool for exploring AAW tactics. The primary objectives of this research are:

- Building a flexible, scalable, expandable, and well documented AAW Analysis Model using DES methodology and the Java Simkit Library.
- Designing, running, and analyzing the outputs of simulation experiments for particular scenarios in a proof of concept demonstration of the potential utility of the model in studying AAW capabilities and tactics. This analysis will also serve as a first step in the validation process of the model.
- Providing the base model for follow on constructive simulations.
- Providing a basis for follow-on research studies, especially for layered defense tactics in naval AAW and formation movement implementation using Java Simkit Library.

D. RESEARCH QUESTIONS

1. Which DES components should be used to build an AAW ship defense model for analyzing AAW capabilities in terms of sensors and weapons?
2. Which analysis methods can be utilized to efficiently conduct simulation analysis?
3. What are the most effective factors in AAW ship design and AAW?

E. METHODOLOGY

Models are used to approximate real systems. This study focuses on models of ships and their sensors and weapons. To model all these objects and their interactions in a DES, some assumptions have been made and some approximations have been used, such as the probability of the kill of an ASM when used against a ship or the probability of kill of a SAM when it is used against an ASM. To obtain the exact inputs and real data for that information was impossible because of the delicate nature of classified data clearance needs. The models studied in this thesis are highly dependent on the data used as input parameters. Because real classified data was not available, the model was run many times over a range of input parameters. As a part of this objective, the effects of these input changes on the outputs are analyzed and documented, as will be explained in the following chapters. The flow of this research is:

1. Determine the simulation model's inputs, output requirements, events, and event details.
2. Identify the simulation event components needed to model and specify the events for these components.
3. Creation of code for the events and components needed for the model.
4. Testing the model in various simple single ship scenarios.
5. Experiment with the model on the convoy scenarios using state-of-the-art design of experiments techniques.
6. Conduct an analysis of the simulation's outputs using suitable statistical tools.

F. SCOPE OF THESIS

This study is limited to an analysis of ship air defense. The ships and the missile properties analyzed approximate real life. All of the data used as input parameters are taken from unclassified open sources. The scope of this thesis is limited to following:

1. Usage of Simkit to create various combat scenarios, objects, and events;
2. Analyzing the effectiveness of sensors, missiles, and combat ships for naval air defense with ship self-defense tactics;
3. The conclusions are based on the simulation results and subject to all model limitations.

G. OUTLINE

1. Introduction

The problem statement and tool (Simkit) is introduced. A brief explanation of the background and methods is made. The scope of the thesis, research questions, and thesis objectives are also in this chapter.

2. Simple Movement and Detection in Discrete Event Simulation Using Simkit Library and Literature Review

The mechanics of DES are mentioned in this chapter. How DES can be used for modeling simple movement and detection using the Simkit library are the body parts of this chapter. The most important point of the chapter is the examination of what studies previously have been done, which is the literature review.

3. Design of AAW Analysis Model

This chapter explains how the AAW Analysis Model is designed, how the model works, what the components of the model are, what the assumptions of the model are, and the strengths and weaknesses of the designed model.

4. Analysis of Model and Results

The fourth chapter presents how the analysis is made, how the experiment is designed, what design methods are used, the measures of effectiveness (MOEs) and measures of performance (MOPs), design points, and the results of the analysis.

5. Conclusions and Future Work

This chapter provides a general overview of how the study has been conducted, the results of the analysis, the insights gained from the study, and what can be done to improve the model.

II. SIMPLE MOVEMENT AND DETECTION

A. DISCRETE EVENT SIMULATION

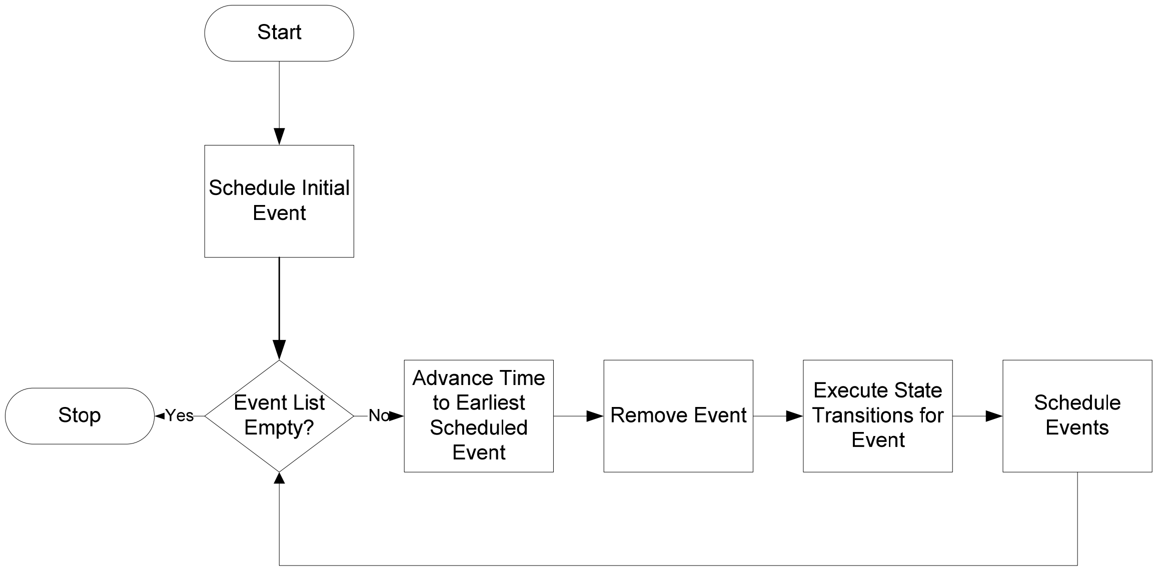
DES is a methodology based on the execution sequence of events at particular times (Law & Kelton, 1991). Each event defined in the simulation occurs at a specific time and an event list keeps track of each event. After an occurrence, an event may or may not schedule another event with a time delay, which can be zero. All these events are stored in a single event list sorted by scheduled time. There are three main elements of DES, they are:

1. States
2. Events
3. Scheduling relationships between events

According to Buss (2011), states can be defined as follows: "[a] state variable in a DES model is one that has a possibility of changing value at least once during any given simulation run. The collection of all state variables for a given DES model should give a complete description of the simulation model at any point in time" (p. 1).

According to Buss (2011), an event can change none, a few, or many state variables, as stated above. Each state transition is a mapping from a model's state space into itself. For each possible state transition, an event is defined.

Buss (2011) explained the next event algorithm by this definition: "The method of time advance in DES models is termed next event. Rather than advancing time in a regular, consistent manner, simulation time moves in typically unequal increments, jumping from the scheduled time of one event to another; thus, the term Next Event" (p. 3). The next event algorithm is shown in Figure 2.



Next Event Algorithm shows how the DES methodology and algorithm works.

Figure 2. Next Event Algorithm. Source: Buss (2011).

According to Buss (2011), a second type of variable is called a simulation parameter. Simulation parameters do not change during a simulation run. The AAW Analysis Model has many parameters and the model is analyzed by systematically varying those parameters.

A simulation run may be terminated in several ways. One of these is ending the simulation run after some specified amount of time has passed. Another way is based on how many times a particular event occurs. For the AAW Analysis Model, the second way is utilized, because the simulation will not always end at a particular time. The simulation run time may differ according to specific scenarios and parameters of objects.

Defining a DES model requires the definition of state variables, parameters, and events by specifying state transitions, assigning a unique name to that event, and defining scheduling relationships between events.

DES and its approach to modeling mechanics have particular advantages compared to those of time stepped models. Especially for combat scenarios generation, even the execution time of the source code may cause drastic differences in outputs. In a time stepped model, even the size of time step may significantly affect the results of a simulation (Al Rowaei, 2011). Particularly, for constructive simulations, the purpose is running the model with different combinations of parameters. Additionally, for each set of parameters we desire to have a moderate amount of replication depending on the size of simulation. In constructive simulations in which our purpose is analysis of outputs, results that are dependent on time steps can be very misleading. On the contrary, in an event based model, this is less of a concern.

B. BASIC EVENT GRAPH MODELING

In the basic DES framework, scheduling events reflects a binary relationship between events being processed and events being scheduled. In other words, knowing which event is being processed allows us to have information about whether an event will be scheduled. Thus, the representation of a framework and the interactions between events is possible by using event graphs in DES (Buss, 2011).

An event graph is comprised of nodes that represent events and directed edges that correspond to the scheduling of other events. Edges may or may not have Boolean conditions that are related to scheduling the next event with an appropriate time delay. In other words, if an edge in an event graph has a condition, the processed event schedules the next event according to the Boolean value of that condition.

The fundamental construct for event graphs is shown in Figure 3. According to Scruben (1983), the construct is interpreted as follows:

> After Event A occurred, it schedules Event B after time delay of *t*, provided condition (i) is true. Condition (i) is evaluated after all the state transitions and necessary calculations are performed in event A. Conventionally, the indication of time delay *t* is made toward the tail of scheduling edge, and the condition related to that scheduling edge is demonstrated just above the wavy line that is in the middle of the scheduling edge. If the time delay is zero (no time delay), then *t* is omitted. In the same manner, if Event B is always

scheduled after the occurrence of Event A, then the edge condition is omitted. So, the basic event graph framework consists of two elements: the event node and the scheduling edge, with options of time delay and edge condition. (p. 983)

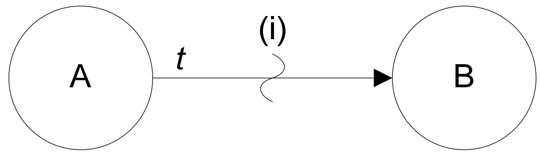


Figure 3. Fundamental Event Graph Construct. Source: Scruben (1983).

Event graphs are critical for making the relationships between events clear and easily understood. A complete event graph model consists of parameters, state variables, event vertices, scheduling edges, and state transition logic.

C. SIMKIT

Simkit (Buss, 2016) is a Java Library designed and developed by Prof. Arnold H. Buss to implement DES. Implementing DES is usually complicated, and commercial software packages typically lack the flexibility that the AAW Analysis Model requires. Specifically, modeling movement, sensing, and weapons interactions are difficult with most commercial software. However, the Simkit library provides a modeler with an intermediate level of knowledge and skills in the Java programming language the ability to model a wide range of applications, including inventory models, queue models, transfer line models, and combat models. Additionally, it has its own built in statistics class. That’s why, for the purposes of this work, Simkit is the most appropriate tool.

D. SIMPLE MOVEMENT AND DETECTION IN DISCRETE EVENT SIMULATION

Entity locations often play the most impactful role in simulation models. However, a time stepped approach has been typically preferred over an event stepped approach to model entity movements, because it has been perceived as more intuitive compared to a movement that is modeled by discrete event simulation. Nevertheless, the time stepped approach has inherent modeling difficulties, artifacts, and limitations. In contrast, modeling movement in a DES approach has many advantages. Buss and Sanchez (2005) define movement in DES as follows:

> At first glance, modeling movement seems to present a challenge to the discrete event approach, since the state of an entity in motion (its location, for instance) is in constant change when an entity is in motion. This difficulty is overcome by the notion of implicit state. An implicit state is one that is not explicitly stored in state variables (instance variables in an object-oriented framework) but rather can be implicitly determined from other state variables. An entity that moves in uniform, linear motion can have its position modeled by implicit state in that its position is not stored as an instance variable but is computed "on demand". The implicit state of position is determined from three explicit state variables: the entity's position when it started to move, the time it started to move, and its velocity vector. (p. 992)

Instead, time is incremented as the events are being processed. When the event happens, first the state variables are changed; then event cancellations, if they occur, are processed; and lastly, further events are scheduled. The state variables do not change in between the events (Buss & Sanchez, 2005). Buss and Sanchez (2005) explain linear uniform motion as follows: "The simplest possible movement is a linear uniform motion. An entity starts its movement at an initial position x at time t_0 and begins moving with velocity v. Its equation of motion is $x + (t-t_0)\,v$, which describes the entities location at time t" (p. 992).

In a DES model, the locations of moving entities are modeled in implicit states, and the movement is uniform and linear, so the location can be easily calculated by dead reckoning using the initial position, velocity vector, and the time it started to move, as mentioned above. The equations of motion explained in the previous paragraph are calculated in base coordinates. Mostly, we need to calculate the location and movement of an entity relative to a particular entity. However, this case is also not hard to represent, because the location of an entity can be calculated with respect to the reference entity and by translating the base coordinate system. After those calculations, the relative velocity is equally trivial as in uniform linear motion. Buss and Sanchez (2005) explain relative velocity as follows:

> The coordinates and velocities of the entities are all in some common base coordinate system, so the motion represented above can be considered absolute motion in the base coordinates. Often it is desirable to consider location and motion relative to some particular entity's coordinates. In that case, the locations and velocities can be represented relative to that entity's coordinates. For most purposes the entities' coordinate systems may be considered to be simply a translation of the base coordinate system. Thus, an entity at position y in base coordinates is at position $y-x$ in the coordinates of an entity located at position x in the base coordinate system. Relative velocity is equally simple for uniform linear motion. Suppose the equations of motion for two entities are given by $x_i + tv_i$, $(i = 1,2)$. Then in the coordinate system of entity 1, the motion of entity 2 is given by $(x_2 - x_1) + \mathrm{t}(v_2 - v_1)$. Thus, relative to the first entity, the motion of the second is uniform and linear with starting position $(x_2 - x_1)$ and velocity $(v_2 - v_1)$. (p. 993)

According to Buss and Sanchez (2005), detection of a moving entity in DES is also possible for modeling radars. The most basic detection is the cookie-cutter sensor. A cookie-cutter sensor detects a target with probability 1.0 when the target enters the range (*R*) of the sensor and loses contact with the target when it gets out of the range of the sensor. So, the question is determining the times that these events are going to occur, which we can call detection and contact loss. The equations of motion for an entity are defined above. According to these equations of motion, the location of a target at a time of detection can be determined as $x + tv$.

Detection occurs when the distance between the target and the sensor is exactly R. We need to determine time t by solving the Equation 3.1 (Buss & Sanchez, 2005):

$$\|x + tv\| = R. \tag{3.1}$$

The detection time is the result of t when the Equation 3.2 is solved (Buss & Sanchez, 2005):

$$\|v\|^2 t^2 + 2(\mathrm{x}\,.\,\mathrm{v})\,\mathrm{t} + \|x\|^2 = R^2 \,. \tag{3.2}$$

Here "." represents the vector inner product and $\| \ \|$ represents the length of the vector. The solutions of Equation 3.2 can be calculated with Equation 3.3 (Buss & Sanchez, 2005):

$$t = -\frac{x.v}{\|v\|^2} \pm \frac{\sqrt{\|v\|^2 (\mathrm{R}^2 - \|x\|^2) + (x.v)^2}}{\|v\|^2} \,. \tag{3.3}$$

This equation has 4 possible results depending on the roots:

1. *Both real positive:* The solutions in Equation 3.3 may both be positive real numbers. In that case, the target starts out of the range of the sensor and is eventually detected. The minimum of the solutions is the detection time and the maximum is the exit time (*A*).

2. *One positive and one negative root:* In this case, the target is already inside the range of the sensor, and the positive root gives the time that the target is going to exit the range (*B*).

3. *Both roots negative:* The target is outside of the sensor range and is moving away from the sensor. At some time in the past, the target passed through the range of the sensor. However, it is never going to enter the sensor range (*C*).

4. *No real roots:* The target never enters the sensor range (*D*).

All of the cases that are explained above can be seen in Figure 4.

The times that are calculated from the expressions can be used to schedule entry and exit times inside the sensor's range. These times are suitable to schedule events in a DES, since events are scheduled with a time delay. These relative times are better for scheduling the event with time delays.

The defined "Enter Range" and "Exit Range" events trigger the "Detection" and "Undetection" events for a sensor. For the cookie-cutter case, the "Detection" and "Undetection" events are scheduled with zero time delay, as expected in the simplest case.

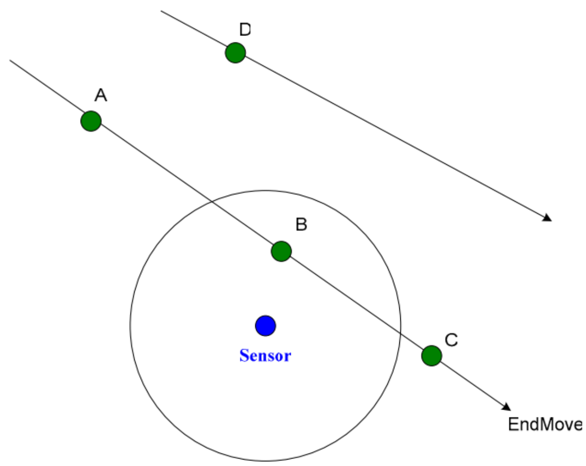


Figure 4. Cookie Cutter Detection: All Possibilities.
Source: Buss and Sanchez (2005).

Another approach that is more complex compared to the cookie cutter approach can be explained as constant rate detection. Since the target is in the range of the sensor, it is going to be detected with a probability of p at every Δt time unit as long as it is inside the sensor's range. Buss and Sanchez (2005) elaborate the constant rate approach as follows:

> Converting this simple approach to a DES application depends on the probability distribution of the time between the "Enter Range" and "Exit Range" events. The detection attempts after the target is inside the range are Bernoulli trials with identical probabilities. Thus, the number *N* detection attempts until the first detection can be explained as a geometric random variable with probability *p*. The time to detection is $N \times \Delta t$, where *N* is a geometric random variable with parameter *p*. The DES formulation requires two parameters, which are Δt and *p*. The Geometric distribution can be simplified as an exponential random variable with mean $\mu = \Delta t / p$. (p. 995)

1. Implementation Approach

The Simkit Library (Buss, 2016) is a Java-based library that supports creating component-based DES models. As mentioned earlier, event graphs describe the state transitions, scheduling, and cancelling relationships between events. Event graphs are directed graphs in which each node specifies an event and state transitions for that event. Each directed arc demonstrates the relationships between events. Connections between components can be made using listener patterns and "LEGO" connections (Buss & Sanchez, 2002).

2. Mover Component

The Mover component is based on the equations of motion that are mentioned earlier. The most important idea about all these implementations is, in DES, events are scheduled at the times when an entity changes its position or state. On the contrary, in a time stepped approach, the entity's state is always updated at each time step (Buss & Sanchez, 2005).

An entity's location cannot be a part of the state of a moving entity, since in DES a state can only be changed when an event occurs (Buss & Sanchez, 2005). Initial conditions remain fixed throughout a given movement. These values can be defined as(x_0, v, t_0), which are initial location, velocity and the starting time for movement, which are the state variables of a basic linear mover. These state variables can be modified or more state variables may be added for more complex movement equations or more complex movers. An event graph for a mover component can be shown as below.

A mover can be defined as any entity that has the capability of movement in a simulation.

Figure 5. Mover Event Graph. Source: Buss and Sanchez (2005).

Maximum speed is among the parameters of a mover component. The most basic command that can be given to a mover component is to move to a determined position at the best speed. This causes the "Start Move" event to be triggered and velocity and movement time is calculated as seen on the event graph in Figure 5 (Buss & Sanchez, 2005). The current location of the event, the current simulation time and the velocity are used as state variables, and they are updated according to events. Velocity is obtained by calculating the vector difference between the current location and destination, normalizing and scaling that unit vector by speed. At the end, the "End Move" event is scheduled with an appropriate time delay. The "End Move" event sets the mover's current location to destination and velocity to zero.

3. Sensor Component

The sensor component has two functional goals. One is keeping the list of detected contacts and the other is holding the parameters needed for the detection algorithm. So, it has only "Detection" and "Undetection" events. The event graph of a sensor component is seen in Figure 6.

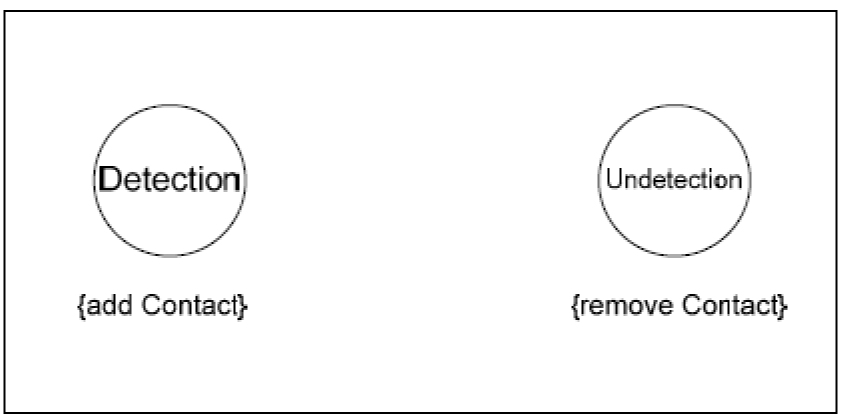


Figure 6. Sensor Event Graph. Source: Buss and Sanchez (2005)

As seen in Figure 6, no scheduling arcs appear in the sensor event graph. Since it does not schedule any event itself, these events are scheduled by another component named the mediator, which is described later. A detection algorithm is implemented by mediators, not by sensors.

Simkit implements a “Sim Event Listener Pattern” (Buss & Sanchez, 2002). Simulation components that are interested in other events which are inside another component are registered as Sim Event Listeners. Whenever a simulation event occurs at the listened component—after making all the state transitions and scheduling the necessary events—listeners are notified. Inside a listener component, events with the same names and arguments are executed.

4. Referee Component

While a target’s range to sensor is more than a sensor’s range, a sensor-target interaction is not possible. However, when the target enters the maximum range of the sensor, detection is possible. Likewise, after a target exits the maximum range of the sensor, interactions between target and sensor no longer matter. Determining when the events “Enter Range” and “Exit Range” are going to occur is the responsibility of a referee component by using the equations of motion that are shown above. A different component has to be used because having the “Ground Truth” data available must not be possible for both the sensor and the mover. A referee keeps a mover-sensor list to detect these movers, listens for “StartMove” and “EndMove” events as seen in Figure 6, and makes necessary calculations according to the equations of motion to determine whether an “Enter Range” or “Exit Range” occurs (Buss & Sanchez, 2005). It also calculates the time delay for “Enter Range” and “Exit Range” events.

In Figure 7, each scheduling edge also has a cancelling edge (Buss & Sanchez, 2005). These cancelling edges are not shown in the event graph to be clearer. The condition (a) is true if the target is outside the sensor range and it is going to enter inside sensor range after t_1 units of time and condition (b) is true if the target is inside the sensor’s range and it is going to exit sensor range after t_2 units of time. Another important aspect not shown in Figure 7 is all “Start Move,” "Enter Range," and "Exit Range" events have two arguments, one of which is Sensor and the other is Mover.

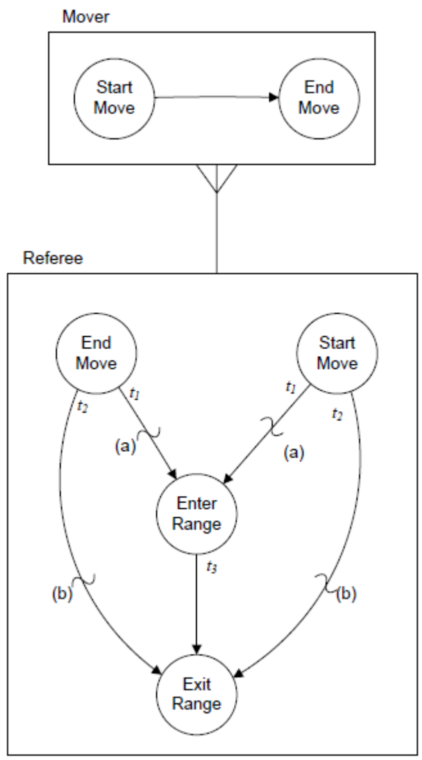
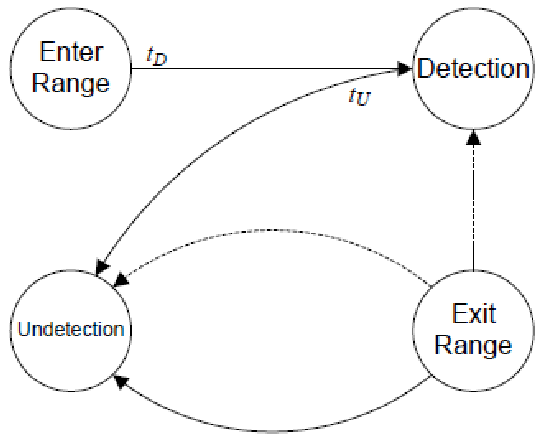


Figure 7. Referee Event Graph Listening to Mover.
Source: Buss and Sanchez (2005).

5. Sensor Mover Mediators

This component's purpose is the same as the referee (Buss & Sanchez, 2005). Because, it is not possible for a mover and sensor to have the "Ground Truth" information, "Detection" and "Undetection" events are scheduled by the mediator component. Usage of this component also helps us have the chance of implementing all kinds of detection algorithms. Adding these events and making these calculations with the referee would force us re-write a different referee class any time a new detection algorithm is added. The event graph for the mediator is shown in Figure 8.



An instance of mediator listens to the referee for “Enter Range” and “Exit Range” events.

Figure 8. Mediator Event Graph. Source: Buss and Sanchez (2005).

An “Enter Range” event calculates the time until detection and schedules a “Detection” event (Buss & Sanchez, 2005). A “Detection” event schedules an “Undetection” event by calculating the time until “Undetection.” An “Exit Range” will be heard by the referee to cancel all the pending “Detection” and “Undetection” events on the event list and schedule an “Undetection” immediately. As it is implemented in Referee, signatures for “EnterRange” and “ExitRange” events are the mover and the sensor. So, these events have parameters and state variables for these objects. Each detection algorithm is going to have a separate Mediator class that implements this. An “EnterRange” event is responsible for scheduling a “Detection” event according to its sensor type using the detection algorithm it is implementing. Detection times are typically calculated using the parameters and state variables that a sensor has. For a Cookie Cutter sensor, the time until detection is always 0.0. For the constant rate detection algorithm described above, the mediator generates random Exponential Random Variables and multiplies this by the mean time to detection to calculate the time to detection and schedule the “Detection” event. More complicated sensor detection models can also be generated and used according to the requirements of the model.

E. LITERATURE REVIEW

A large amount of work, studies, papers, and theses relate to AAW. However, most of them are not related to the specific task of building a Discrete Event Simulation Tool for a complete analysis of AAW. Among all of them, the ones highlighted below contributed significantly toward that goal.

Kulac built an analysis tool whose purpose is “to make a comparative analysis of active and passive sensors in AAW defense using DES components” (Kulac, 1999). He developed an analysis tool to measure to measure the effectiveness of infrared and radar Sensors in AAW. He used a component-based simulation approach similar to the one used in this report. His tool was scalable and flexible. Additionally, he statistically analyzed different MOEs. However, in his study, he only focused on the sensors and two primary classes of weapons. These were ASMs and SAMs. Nevertheless, modern warships are such complex systems that their AAW capabilities are not limited to that extent. Inclusion of a layered defense capability for a warship in AAW is crucial. The other issue his report neglected was formation movement and defense tactics at a naval task group level. He mostly focused on a single ship’s self-defense. That’s why his model’s fidelity was limited. On the other hand, he provided a good study given that the Simkit library was so new at that time. Since then, there have been many improvements to Simkit.

Aydin modeled the screen dispositions of naval task forces (Aydin, 2000). He also built a tool and graphical user interface for ship defense in convoy operations. In his model, he used a Disposition Mission Model (DMM) to perform an effective defensive disposition from a task force. He focused on the Graphical User Interface (GUI) to provide a user friendly environment for analyzing new tactics and formations. He modeled two dispositions for AAW defense against particular types of ASM missiles. These dispositions were Screen Disposition and Disposition 2W. He studied the effects of the disposition of naval units on the defense of a High Value Unit in a convoy operation. In his model, he spawned ASM missiles from particular threat sectors toward a convoy and checked whether the convoy had succeeded in the defense of HVU. Then, he made a logistic regression to check which factors were effective. He considered the ranges of missiles, the axis of movement, the position of the threat sector, the number of ships, the number missiles, and the number of ships on the threat sector. Nevertheless, his model lacked the fidelity of the engaging unit that sent the ASMs to the convoy and ship's layered defense models.

Turan developed a simulation that provided "suitable Operations Research analytical techniques and tools to aid decision authorities in the Ship Self Air Defense (SSAD) system selection process" (Turan, 2002). He used DES techniques and implemented them in the Java Programming Language and Modkit. Then, he used his simulation to analyze two different SSAD systems and firing policies. He defined the key parameters as number of trackers, SAM inventory levels, and slew delay. He made a comparative analysis of Shoot-Look-Shoot and Shoot-Shoot-Look policies in fire control systems and Active and Semi-Active ship self-defense systems. As a result of the success of his SSAD simulation, using the success criteria as no leak to the ship, he made recommendations for further component additions and modifications. In his model, he did not take the layered defense policy into consideration and did not model the gun and CIWS of a ship. These weapon systems play a crucial role in AAW. Furthermore, he only focused on a single ship's self-defense policy.

Townsend—in his report about the defense of Naval Task Forces from an ASM attack—developed an analysis tool called ASM Defense Model (Townsend, 1999). The model allowed for an analysis of entire task force by modeling ASMs' target selection and escort ships' protection of HVU by defensive fire. He studied an effective screen design and defensive firing policy. He created a mathematical library to solve various equations of motion. The model he built could also evaluate missile attacks from different angles and the impact of a decoy if it were developed. He suggested this subject for a future study. He also used the Simkit and Modkit libraries in the Java Programming Language. However, his work was solely focused on ASM raids, and he also did not consider including a layered defense policy. The only objects he had in the simulation were ASMs, ships and SAMs.

No studies have been found of a complete analysis of complex AAW scenarios with formation movement models and ship layered defense models with SAMs, guns, and CIWS, so no tools comprehensively assess AAW in a naval operational area. Because of the insufficiencies and artifacts of the tools that are built, a new tool for a complete analysis of AAW was required. A statistical study of model output with many factors explored is conducted, as will be seen in Chapter IV.

III. DESIGN OF AAW ANALYSIS MODEL

This chapter describes the AAW Analysis Model, which is a stochastic DES model. The AAW Analysis Model was developed to investigate the most effective factors for the protection of a High Value Asset in convoy operations. Two types of ships are modeled inside the AAW analysis model: a High Value Unit (HVU) and frigates. Analysis inside the AAW Analysis Model is based on a primary scenario of the HVU's protection by friendly frigates against enemy frigates. The HVU and all the frigates have specific starting locations according to the scale of the Simkit smd library, which uses a two dimensional Cartesian coordinate system. The HVU has a predefined path for each scenario and the simulation run will terminate according to the given condition of either the HVU's destruction or the HVU reaching the last waypoint of its predefined path. Each frigate that represents enemy ships also has predefined paths. These enemy frigates patrol on those predefined paths. Each friendly frigate of the HVU has a starting position close to the HVU. Each of these frigates takes their positions according to the relative offset angle and offset distance in the screen formation. Friendly frigates protecting the HVU move by keeping their relative distance and offset angle from the HVU. Their goal is the protection of the HVU by either engaging enemy ships or destroying the ASMs that are sent from enemy ships against either themselves or the HVU. One possible scenario consists of three Blue frigates, one HVU, and two enemy ships, as seen in Figure 9.

The threat axis at the starting conditions of the simulation is seen in the Figure 9. However, that axis is subject to change due to the movement of units. Actually, there may be a 360 degrees threat against the HVU and Blue ships. Incoming threats are eliminated by ships according to a layered defense policy. After the detection of an incoming air threat (particularly a missile), soft kill methods like Electronic Warfare and decoys are conducted when the threat reaches a specified distance from the units. Then, AAW defense ships first engage with their surface-to-air missiles (SAM) to eliminate the incoming anti-ship missiles (ASMs). If some ASMs are not eliminated, then the defensive ships engage with their guns, and if still not eliminated they use a last decoy to attempt to deceive the incoming ASMs, and finally they engage with their CIWS. Air threats may also stem from airplanes. However, neither airplanes nor soft kill methods are currently modeled in the AAW Analysis Model. Thus, air threats that are detected and engaged with a layered defense policy can only be ASMs, and they will be destroyed by only the hard kill methods that are mentioned above in the layered defense tactics. Thus, ASMs can only be launched from ships.

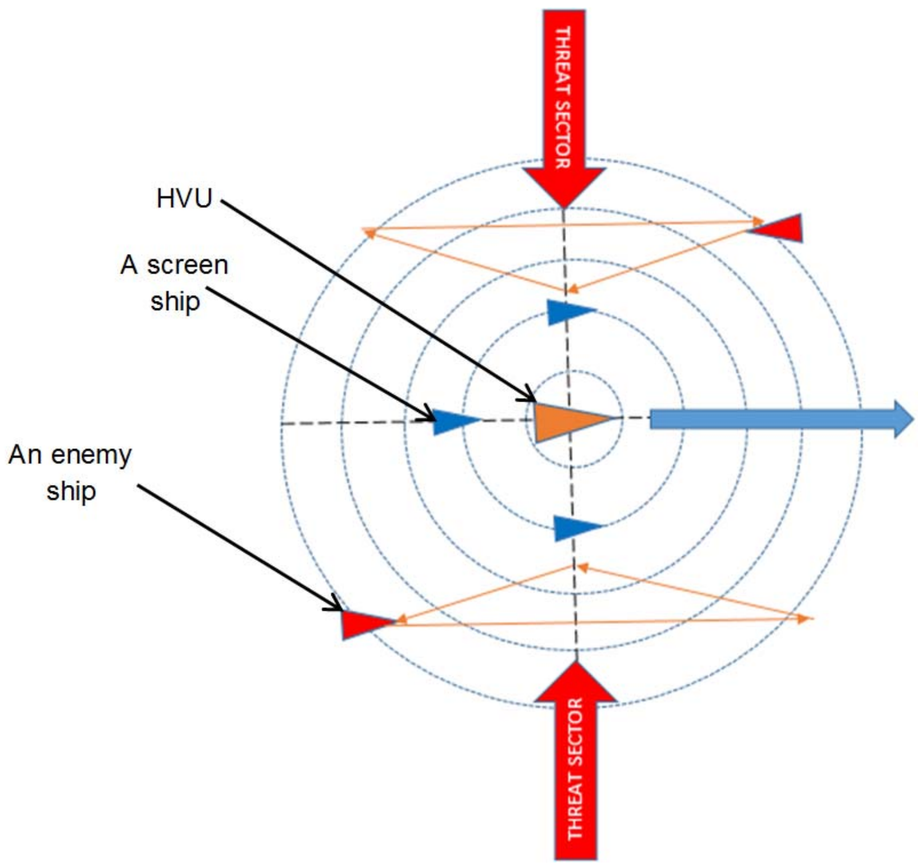


Figure 9. Possible Scenario for AAW Analysis Model.

In Figure 10, concentric circles represent each ship's sensor ranges. A blue circle is a ship's surveillance sensor range. Inside this range, an entity can be detected and classified by ships. A red circle represents SAM engagement sensor range for a ship. Inside that range, ships engage an incoming ASM with a SAM. A yellow circle represents gun engagement range. Inside that range, ships engage their targets with its gun. A black circle represents CIWS engagement sensor. Inside that range, ships engage their targets with CIWS. Red ships (enemy ships), which are the ones at the left most top corner and right most bottom corner, and screen ships, which are located around the HVU in the middle of the figure, have a surveillance sensor, SAM engagement sensor, gun engagement sensor, and CIWS engagement sensor. HVU has only a surveillance sensor and CIWS engagement sensor. An orange square represents an ASM; a blue square represents a SAM; a red square represents a gun round, and a black square represents a CIWS round.

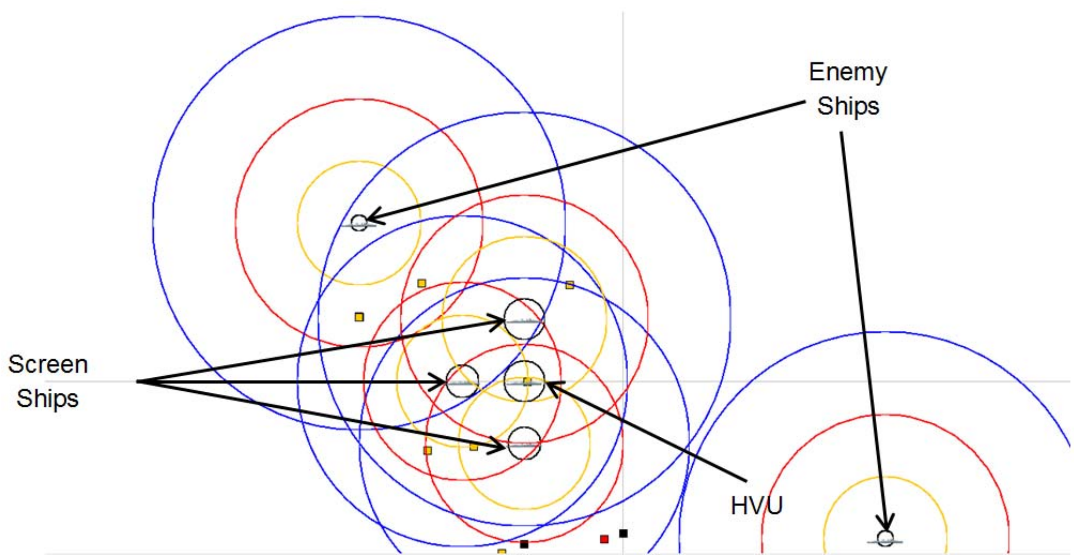


Figure 10. Demonstration of the AAW Analysis Model.

Ships may engage each other with their ASMs and guns if they are inside ranges of their corresponding weapons. A ship may be hit by an ASM if that particular one leaks from the layered defense and successfully hits a ship. A ship being hit by a gun depends only on a successful hit, with two conditions:

- Sending the weapon to a location that is close enough to cause an impact for a ship.
- Successfully damaging the ship.

An ASM can be eliminated if a SAM, gun round or CIWS round gets close enough for impact and successfully damages the ASM. So, inside AAW Analysis Model, five types of engagements may occur among weapons or ships:

- ASM (launched by enemy ship)-ship engagement
- Gun round (fired by enemy ship)-ship engagement
- ASM-SAM (launched by opposing units) engagement
- ASM-gun round (launched/fired by opposing units) engagement
- ASM-CIWS (launched/fired by opposing units) engagement

All the actions a ship can take depend on its capabilities. In the AAW Analysis Model, a ship has the capability to detect enemy units or the ASMs launched from enemy units with its surveillance radar, classifying them correctly, and engaging them with appropriate weapons. A frigate can have ASMs, SAMs, guns and CIWS in the AAW Analysis Model. However, even though it may be modified later, currently an HVU can only have a CIWS for the scenarios to be analyzed. SAMs and CIWS are not considered to be threats against ships; rather, they are primarily defensive weapons.

The ship self-defense and attack system that is developed inside the AAW Analysis Model consists of the components that are listed as follows:

1. Ship
2. ASM
3. SAM
4. Missile mover manager
5. Follower mover manager
6. HVU mover manager
7. Gun round

8. Round mover manager
9. Ship surveillance sensor
10. Ship sensor for SAM engagement
11. Ship sensor for Gun engagement
12. Ship sensor for CIWS engagement
13. Contact
14. Policy
15. Adjudicator

Among those components, the ship surveillance sensor, the ASM, the SAM, the gun round, the contact, and the adjudicator are stochastic ones. The ship surveillance sensor is a constant rate sensor. A “Detection” event inside the ship surveillance sensor is scheduled according to an exponential distribution whose rate is determined by user. The ASM, the SAM, the gun round components have their own damage functions. Their damage functions return damage amounts if an ASM, a SAM, or a gun round successfully hit their targets. The damage amounts come from a truncated normal distribution. The parameters of those damage functions are among parameters of the AAW Analysis Model. A contact component creates distortion for each sensor’s detection. This distortion comes from a rotated bivariate normal distribution whose parameters are also among the AAW Analysis Model factors. The adjudicator component has user defined probability distributions for each type of engagement.

A. SHIP

A ship is the main object of the simulation. A ship object is designed by extending the Basic Linear Mover class in Simkit. The ship object is designed such that any type of ship can be instantiated using that single class. It has its own methods for launching its missiles, or firing its guns, if it has any. For instance, if ship is an HVU, such as an aircraft carrier or a tanker, it can only have CIWS. On the other hand, a ship can be instantiated with any type of weapon and sensor combination provided we have sufficient data for sensor and weapon ranges, missile speeds, gun or missile inter shoot/launch delays, or even the detection rate of the sensor. All kinds of sensors are parts of the ship object. A ship listens to its sensors for any kind of detection.

A ship object's Unified Modeling Language (UML) diagram and event graph are seen in Figure 13 and Figure 11, respectively. For the simulation to execute properly, all the "Sim Event Listener Patterns" should be instantiated before each simulation run. General "Sim Event Listener Pattern" for the AAW Analysis Model is shown in Figure 12. If an object is destroyed, the sim event listeners from it are removed. All the objects that are listening to it stop listening.

B. ANTI-SHIP MISSILE

ASM is the component with the properties and state variables shown in the UML diagram (see Figure 14). It is designed as an object in the simulation. It has alive, engaged, and destroyed states. After it has been instantiated and sent toward the target, it enters an engaged state. After impact, it enters a destroyed state, regardless of the result of the impact. The ASM uses the missile mover manager for its movements. When the ASM is launched, it gets its target as a parameter during instantiation and it attacks toward the target.

ASM is a munition type in the AAW Analysis Model. All ASMs are instantiated inside an ASM pool list for each ship at the very beginning of simulation runs. These ASMs inside the ASM pool are used for all simulation runs. In each simulation run, ASMs are popped from the ASM pool list and inserted in a launched ASM list in "Launch ASM" event when the ships engage with theirs ASMs. After the simulation is reset for another run, all ASMs in both the ASM pool list and the launched ASM list are reset. Then, launched ASMs are popped from launched ASM list and inserted back to the ASM pool list of that ship. In that way, the ASM pool list and the launched ASM list return to the original states. This methodology is used to optimize the code and execution speed. Since the ASMs are dynamically used at each ASM engagement, instantiating them at each engagement without using the pooling methodology causes an unused memory leak and slows down the simulation. The ASM component event graph is shown in Figure 15.

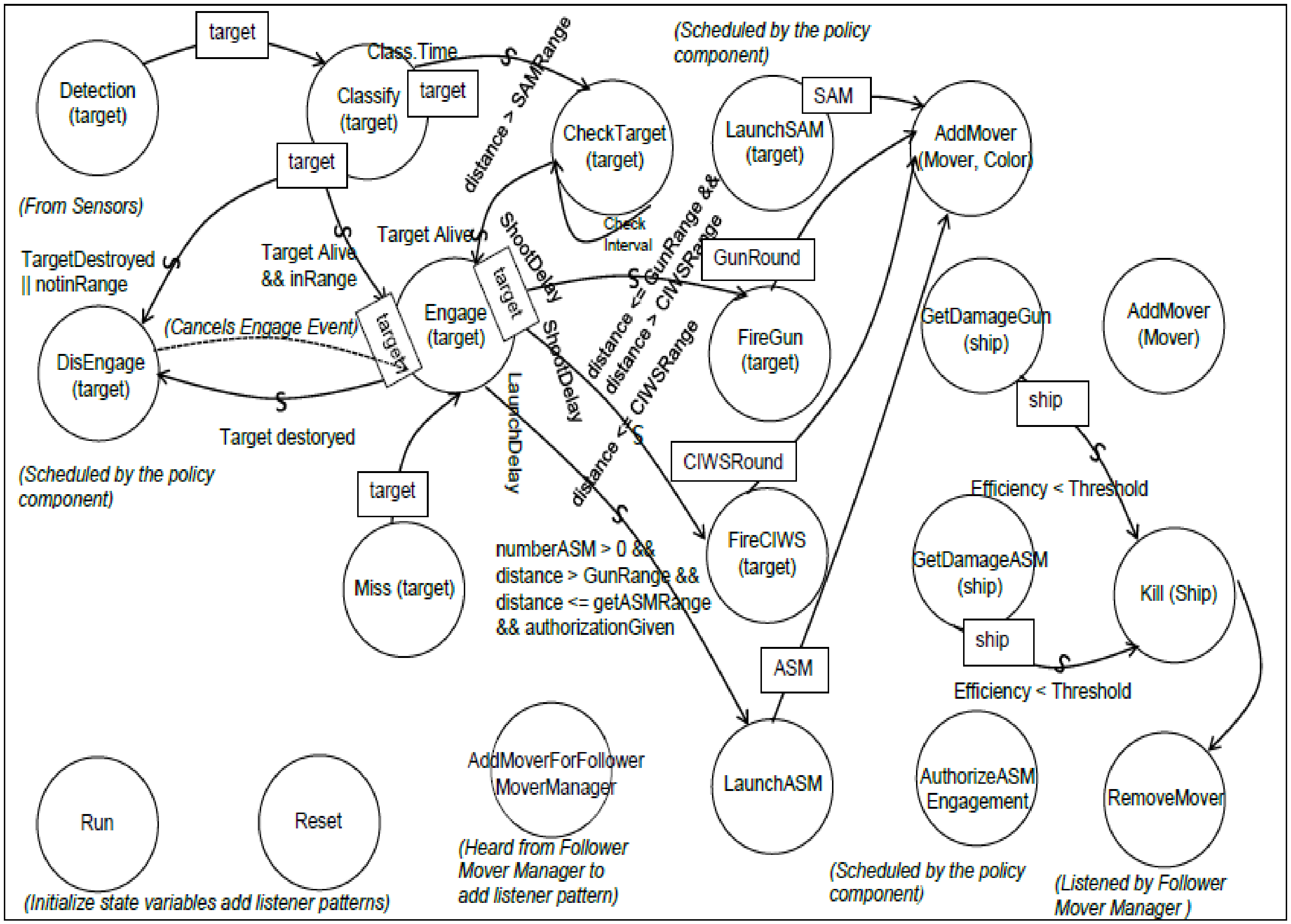


Figure 11. Ship Event Graph.

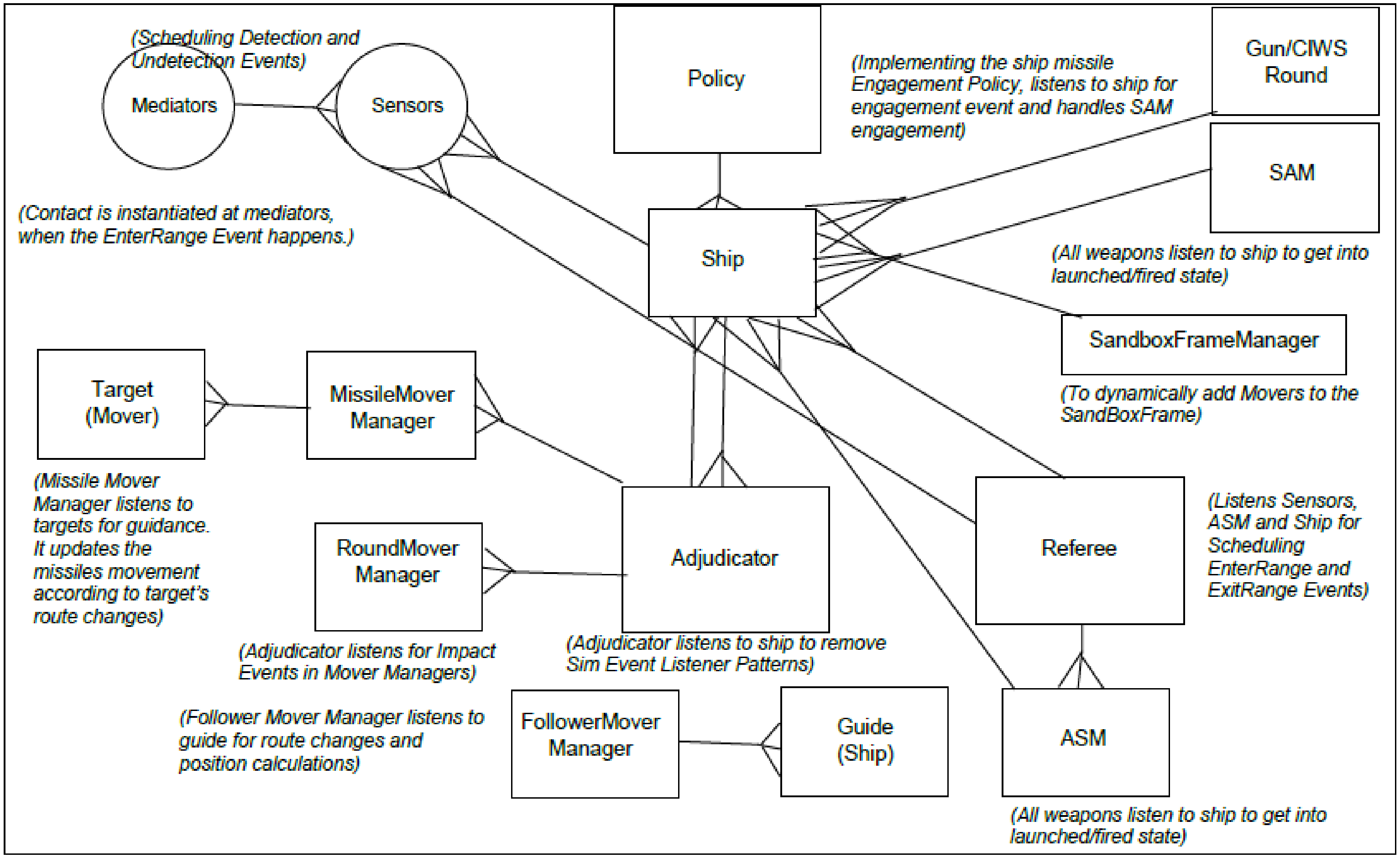


Figure 12. General Sim Event Listener Pattern for AAW Analysis Model.

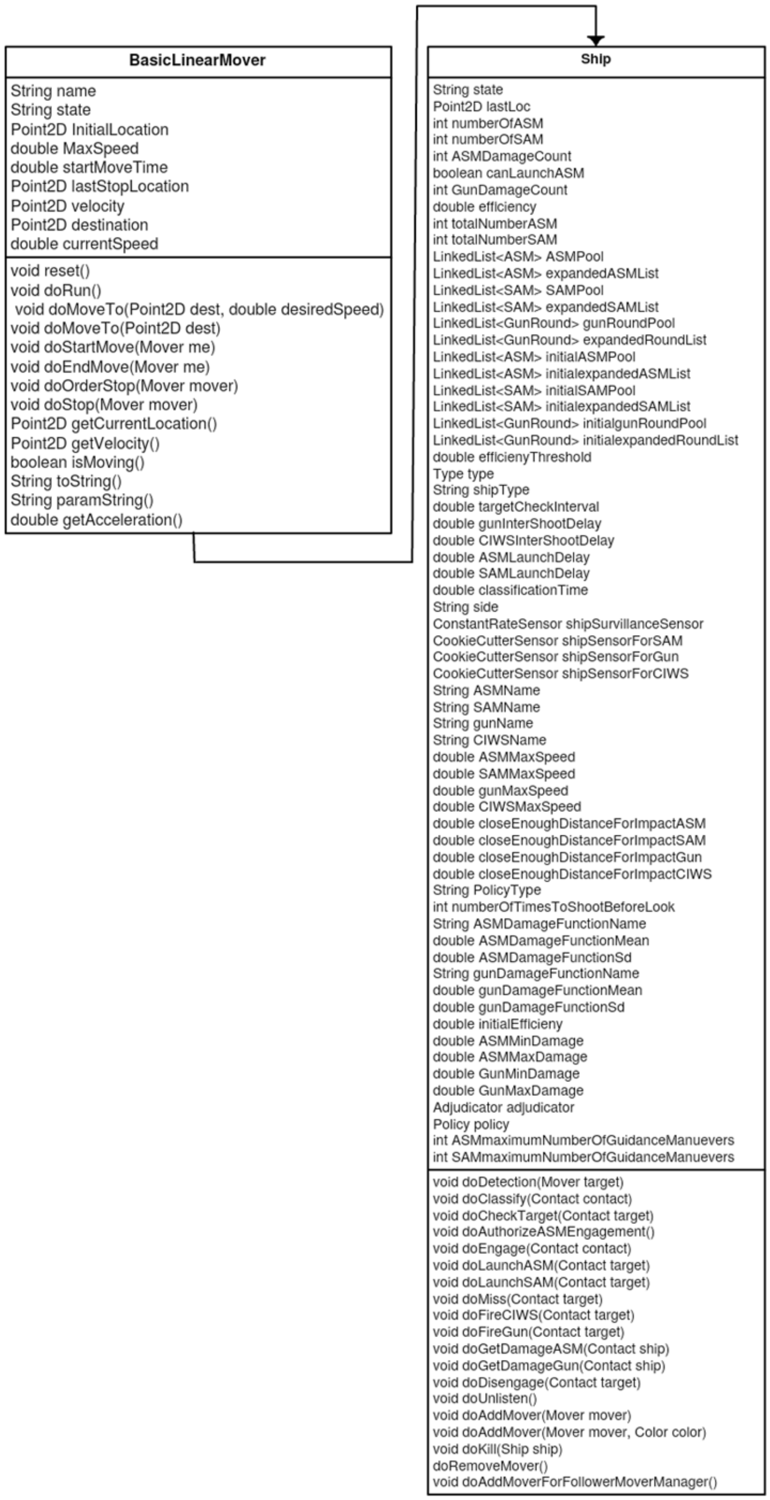


Figure 13. Ship UML Diagram.

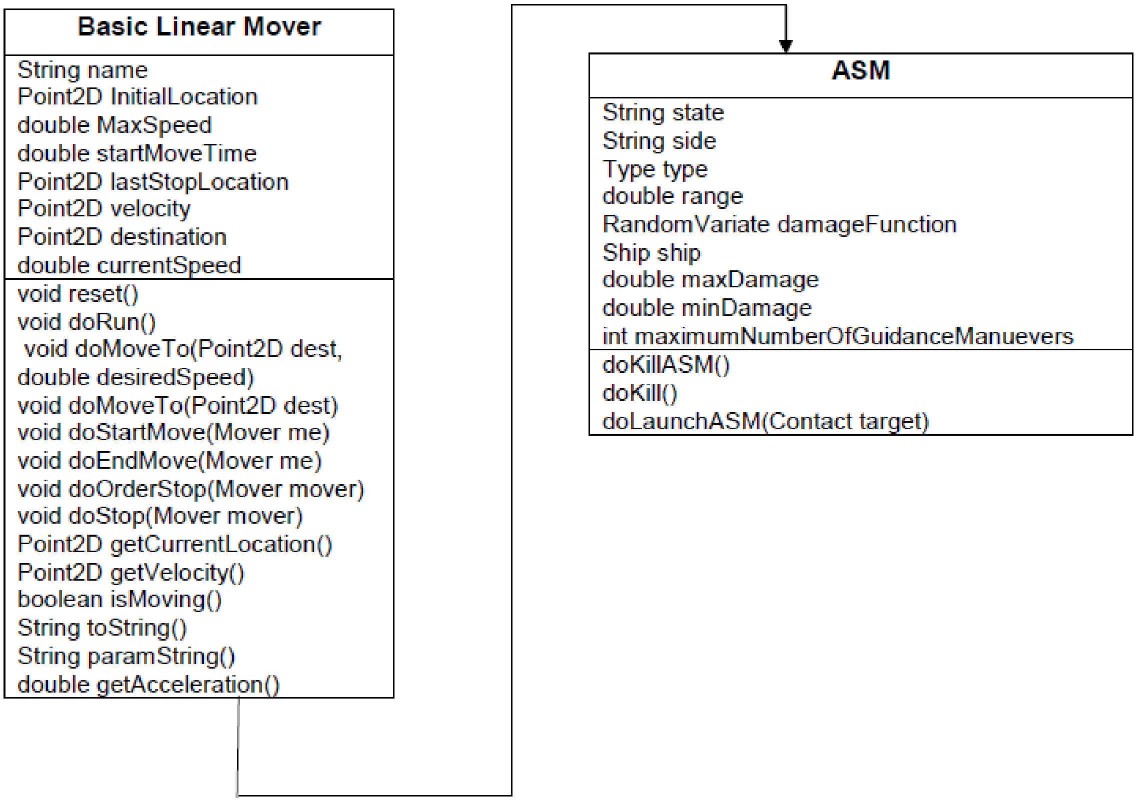


Figure 14. ASM Component UML Diagram.

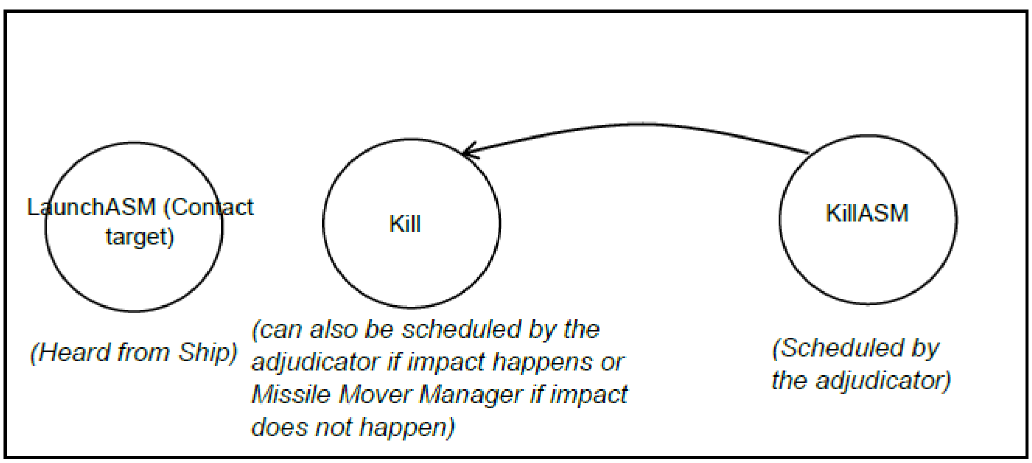


Figure 15. ASM Component Event Graph.

C. SURFACE-TO-AIR MISSILE

SAM is the component that is used at one of the ship self-defense layers, when triggered by the ship sensor for SAM engagement. It uses the missile mover manager as an ASM. SAM is an object similar to ASM. It moves according to the target that is defined at its mover manager at instantiation. It is instantiated dynamically with pooling methodology at the engagement event, like an ASM. SAMs are launched and reset with the same methodology used in ASM engagements. The SAM component UML diagram is shown in Figure 16, and the SAM component UML diagram is shown in Figure 17.

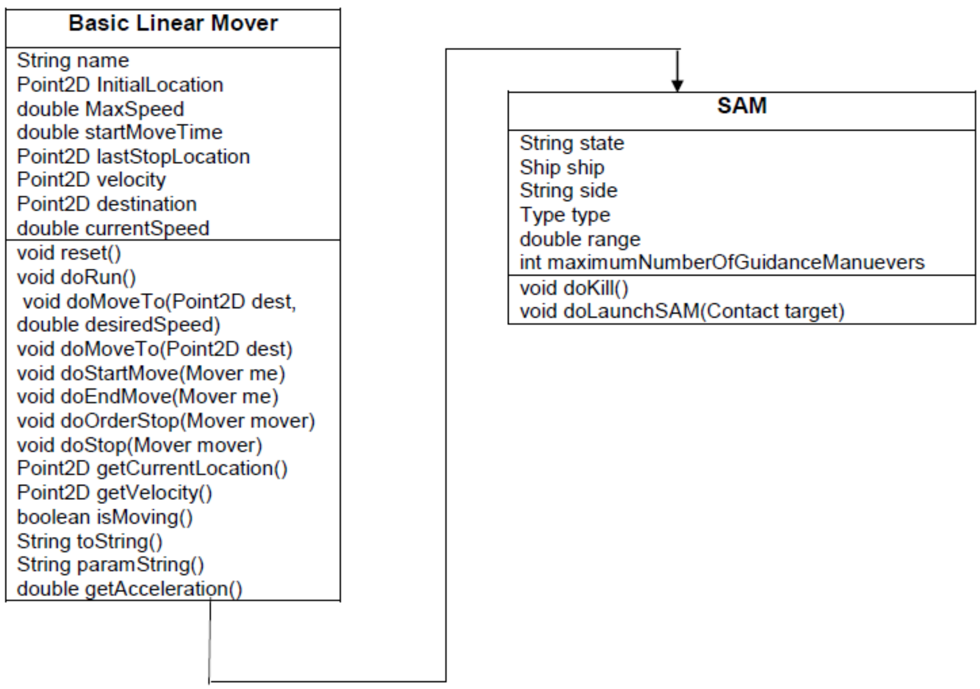


Figure 16. SAM Component UML Diagram.

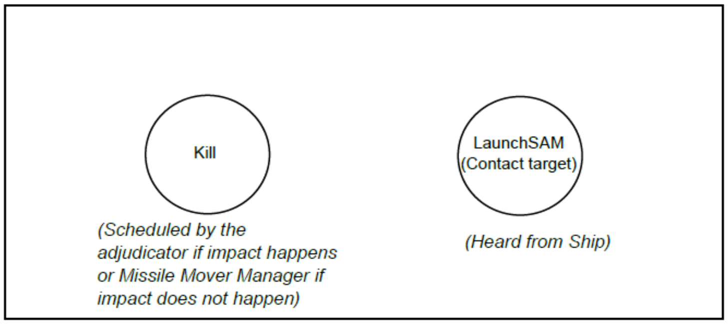


Figure 17. SAM Component Event Graph.

D. MISSILE MOVER MANAGER

The missile mover manager is a special mover manager that is designed for the movement of guided missiles. As seen in its event graph, the missile mover manager calculates a new interception when the route of its target changes. So, it guides its mover toward its target. It has a parameter that defines the impact distance. The missile mover manager UML diagram is shown in Figure 18, and the missile mover manager event graph is shown in Figure 19.

MissileMoverManager (extends SimEntityBase)
Point2D nextIntercept int numberOfGuidanceManeuvers boolean startOn Mover mover Contact target double closeEnoughForImpact int maximumNumberOfGuidanceManuevers
doStartMove(Mover target) void doStart() void doIntercept() void doMoveTo(Point2D destination) void doEndMove(BasicLinearMover mover) void doImpact(MissileMoverManager ASMMoverManager, ASM asm, Contact ship) void doImpact(MissileMoverManager SAMMoverManager, SAM sam, Contact asm) void doUnlisten() void doStop() void doOrderStop(Mover mover)

Figure 18. Missile Mover Manager UML Diagram.

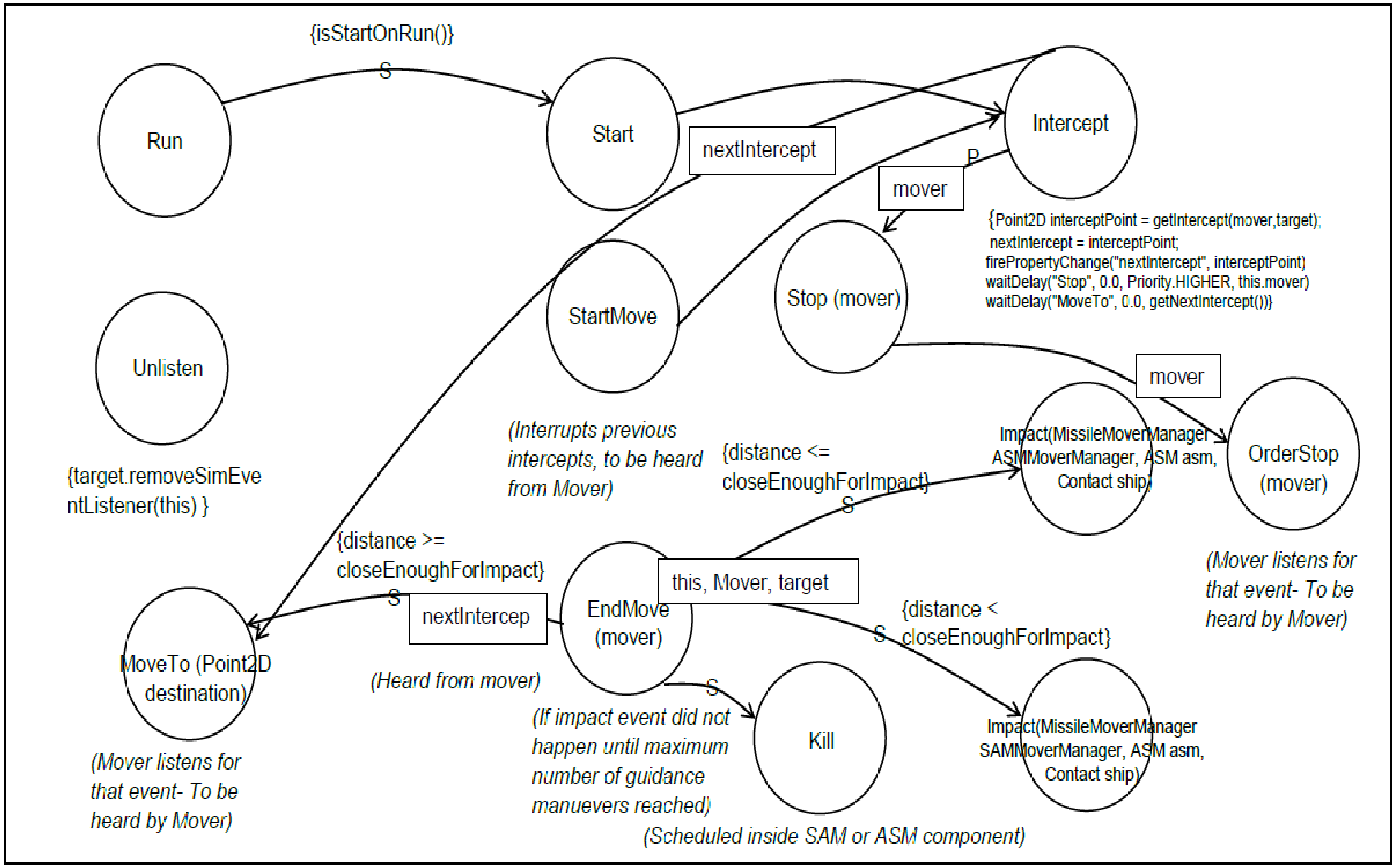


Figure 19. Missile Mover Manager Event Graph.

E. FOLLOWER MOVER MANAGER

The follower mover manager is the mover manager responsible for movement of the screen ships according to a guide. The follower mover manager calculates the position of a screen ship according to a relative distance and angle from its guide by putting the guide at the center of the coordinate system. After making the calculations, the follower mover manager orders the movement to that specified location at maximum speed for its mover (a screen ship). The mover takes its relative position at its best possible speed. Then, the follower mover manager makes its mover (a screen ship) adjust its speed according to the guide and protect its relative position. Thus, the follower mover manager also listens and checks for the guide ship's movements to order its mover to a new destination. The follower mover manager UML diagram is shown in Figure 20, and the follower mover manager event graph is shown in Figure 21.

FollowerMoverManager (extends SimEntityBase)
boolean startOn Mover mover Mover guide double offsetDistance double offsetAngle Point2D nextLocationToMove
void doStartMove(Mover target) void doCalculateNextLocationToMove() void doMoveTo(Point2D destination) void doEndMove(BasicLinearMover mover) void doStop() void doOrderStop(Mover mover)

Figure 20. Follower Mover Manager UML Diagram.

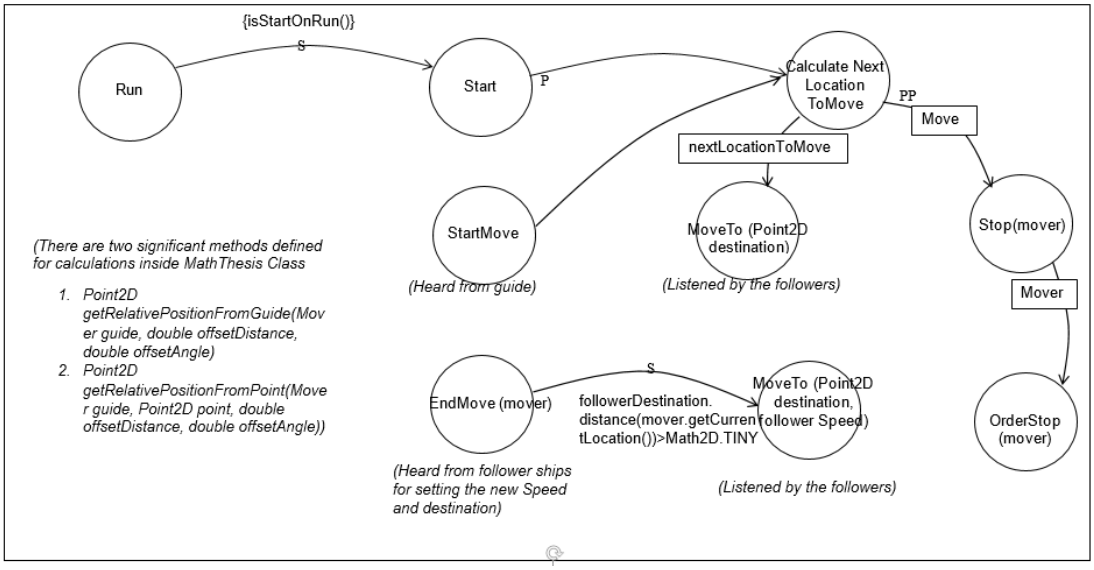


Figure 21. Follower Mover Manager Event Graph.

F. HIGH VALUE UNIT MOVER MANAGER

The HVU is the guide of the disposition movement according to the AAW Analysis Model. So, the screen ships adjust movements according to the HVU by listening to it. This is the real phenomenon, and is true for real operations. The HVU mover manager is a mover manager that makes the HVU move on the assigned path. It has an additional method to finish the simulation checking if the HVU reached its last stop location assigned at the beginning of simulation with a database file. If the HVU reaches that location, the simulation ends with success.

G. GUN ROUND

The gun round component for the ship is designed for the gun and CIWS engagement layer of the ship self-defense. Like SAM and ASM, it is an object and it is instantiated from the gun round pool list dynamically at the engagement step of gun and CIWS. The differences of gun and CIWS rounds are their speeds and probability of target kills. They are fired and reset with the same methodology that is used in ASM and SAM engagements. The gun round event graph and UML diagram are shown in Figures 22 and 23, respectively.

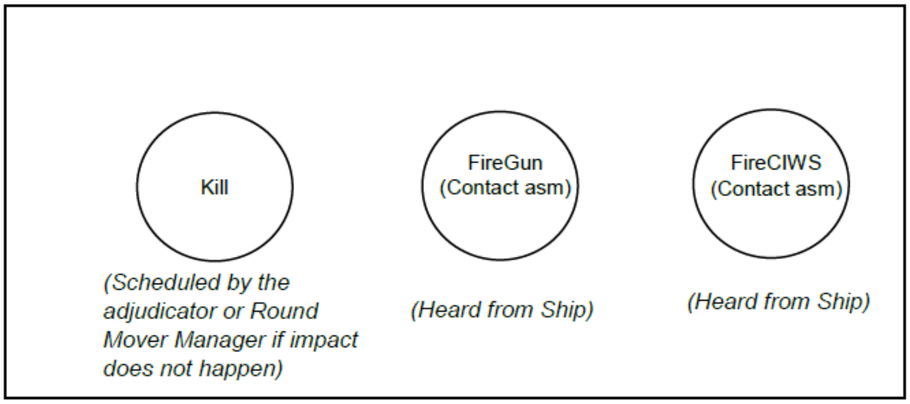


Figure 22. Gun Round Event Graph.

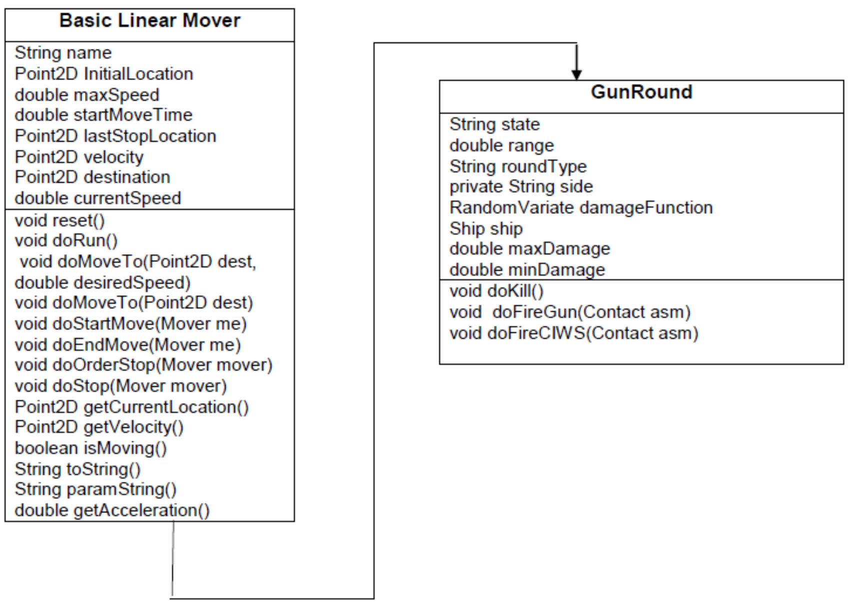


Figure 23. Gun Round UML Diagram.

H. ROUND MOVER MANAGER

The round mover manager is the component used to move the gun and CIWS rounds. Unlike missile mover manager, it is not a mover manager that makes its mover adjust toward its target. The intercept of the target and the round is calculated at the beginning of the engagement process and makes its mover go toward the target. It has a parameter that defines the impact distance like a missile mover manager. The round mover manager UML diagram is shown in Figure 24 and the round mover manager event graph is shown in Figure 25.

RoundMoverManager (extends SimEntityBase)
boolean startOn Mover mover Contact target double closeEnoughForImpact
doStartMove(Mover target) void doStart() void doIntercept() void doMoveTo(Point2D destination) void doEndMove(BasicLinearMover mover) void doImpact(RoundMoverManager roundMoverManager, GunRound gunRound, Contact target) void doUnlisten() void doStop() void doOrderStop(Mover mover)

Figure 24. Round Mover Manager UML Diagram.

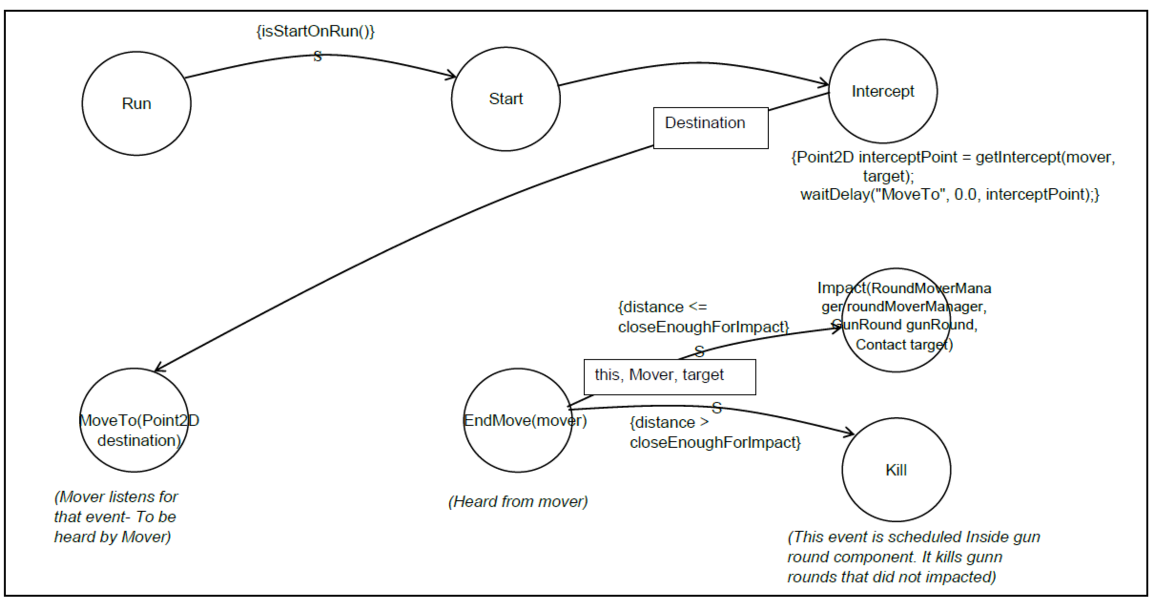


Figure 25. Round Mover Manager Event Graph.

I. SHIP SURVEILLANCE SENSOR

The ship surveillance sensor is the primary sensor of the ship. It is a constant rate sensor. It has a rate of detection and a time delay to detect the target according to the rate of detection. The ship listens to its surveillance sensor for detection and classification events. It basically readies for engagement according to its target type after classification. For instance, if the target is classified as a ship, then the ship starts to engage with an ASM, or if it is defined as ASM, then the ship waits for the target to get inside its SAM engagement range and starts its layered defense.

J. SHIP SENSOR FOR SAM ENGAGEMENT

The ship sensor for SAM engagement is a cookie cutter sensor mounted on the ship to serve as a trigger for SAM engagement. After the target is detected by this sensor, the ship starts its engagement with a SAM. It is defined as a cookie cutter sensor because of its functionality.

K. SHIP SENSOR FOR GUN ENGAGEMENT

The ship sensor for gun engagement is a cookie cutter sensor mounted on the ship to serve as a trigger for gun engagement as part of a layered defense policy. After the target is detected by this sensor, the ship starts its engagement with its gun. It is defined as a cookie cutter sensor because of its functionality.

L. SHIP SENSOR FOR CIWS ENGAGEMENT

The ship sensor for CIWS engagement is a cookie cutter sensor mounted on the ship to serve a trigger for CIWS engagement as part of a layered defense policy. After the target is detected by this sensor, the ship starts its engagement with CIWS. It is defined as a cookie cutter sensor because of its functionality.

M. CONTACT

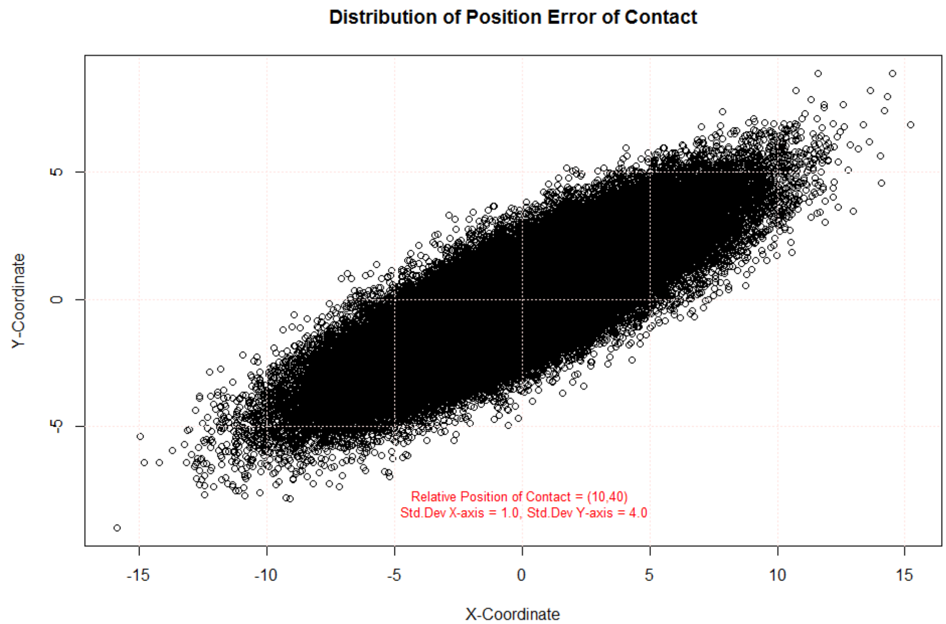
The contact component is a component that serves a mediator between the ship and the sensor after the enter range event of the sensor. In the enter range event of the sensor, contact is instantiated with the sensor that detected the real object and the real object that is detected. The contact is passed as a parameter to the detection event instead of the real mover object. It has limited information about its real mover. It has a position distortion that is created according to a rotated bivariate normal distribution that has a standard deviation according to the detected real object's relative position and its distance. This component is used to increase the realism and fidelity of the simulation, and it is essential for combat simulations, see Lucas (2000). For each of the sensors mentioned above, contact is provided with the sensors' distortion factors and parameters. The contact component UML diagram and source code is shown in Figure 26 and an instance of distortion created by the contact component is shown in Figure 27.

Contact (extends SimEntityBase)
String state Mover realMover Sensor sensor double noiseOffsetOnXAxis double noiseOffsetOnYAxis double noiseFactorOnXAxis double noiseFactorOnYAxis

Point2D getCurrentLocation(){
Point2D realPositionMover = getRealMover().getCurrentLocation();
Point2D realPositionSensor = getSensor().getCurrentLocation();
Point2D relativePosition = new Point2D.Double (realPositionSensor.getX()-realPositionMover.getX(),realPositionSensor.getY()-
realPositionMover.getY());
double distance = realPositionMover.distance(realPositionSensor);
double angle = Math.atan2(relativePosition.getX(), relativePosition.getY());
RandomVector rv = RandomVectorFactory.getInstance("RotatedBivariateNormal", new Object[]{getNoiseOffsetOnXAxis()/1852,
getNoiseOffsetOnYAxis()/1852,
getNoiseFactoronXAxis()/1852*distance, getNoiseFactoronYAxis()/1852*distance, angle});
double [] noise = rv.generate();
Point2D locWithNoise = new Point2D.Double(realPositionMover.getX() + noise[0], realPositionMover.getY() + noise[1]);
 	
return locWithNoise;}

Inside the diagram we can see the source code that is used to generate random locations to simulate distortion of a sensor.

Figure 26. Contact Component UML Diagram and Source Code.

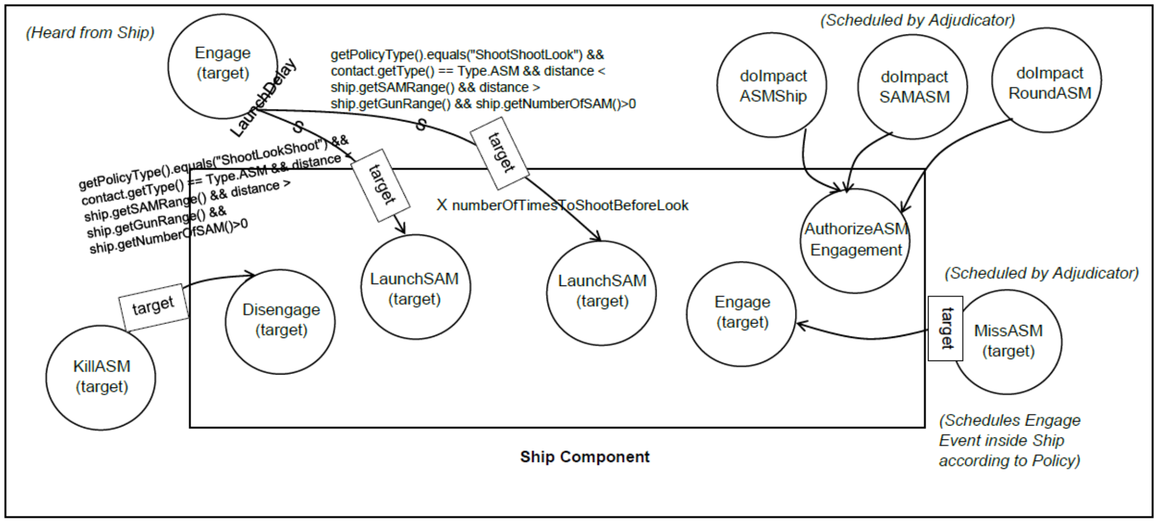


This distortion is created by Contact Component for 10000 runs.

Figure 27. An Instance of Distortion.

N. POLICY

The policy is the component that controls ASM and SAM engagement processes for the ship. The ship may have two different policies for the engagement of SAM. One of them is Shoot-Shoot-Look and the other is Shoot-Look-Shoot. These policies are implemented according to the policy component. If the Shoot-Shoot-Look policy is in force, the ship sends a certain number of missiles. The number of missiles sent is a parameter of the policy component. The ship checks whether the target has been destroyed and if it has not been destroyed then it sends another salvo. If the Shoot-Look-Shoot policy is in force, then the ship sends one missile before checking if the target has been destroyed, and if it has not been destroyed, sends another missile. These policies show the tradeoff between SAM inventory management and survivability. The policy component event graph and UML diagram are shown in Figures 28 and 29, respectively.



In the event graph, the interaction of policy and the ship component is also shown.

Figure 28. Policy Component Event Graph.

Policy (extends SimEntityBase)
Ship ship String PolicyType int numberOfTimesToShootBeforeLook int numberASMlaunched int missCounterForShootShootLook
void doEngage(Contact contact) void doImpactASMShip() void doKillASM(Contact target) void doImpactSAMASM() void doMissASM(Contact target) void doImpactRoundASM()

Figure 29. Policy Component UML Diagram.

O. ADJUDICATOR

The adjudicator component serves as a decision module defined as part of each ship that is instantiated in the simulation. Each adjudicator for allied ships has the same parameters. The Red and Blue units may have different adjudicator parameters. It decides whether the target is destroyed or damaged after any particular impact event, according to a user-defined probability distribution. An impact event happens if a mover gets close enough to its target. For instance, these movers may be ASM, SAM, and gun. Each ship's adjudicator makes judgments about its own ship and any interaction the ship or its weapons may have. It listens to mover managers of that ship's weapons for impact events that are defined in mover managers of launched or fired weapons. It also listens to its own ship for any kill event. If a kill event happens for its own ship, it stops listening to the ship and removes the event listener pattern between the ship and that ship's policy component. Each ship listens to its adjudicator for miss events to reengage targets. In accordance with its adjudicator, the ship engages its target, if its target is not destroyed. The adjudicator component event graph is shown in Figure 30 and the adjudicator component UML diagram is shown in Figure 31.

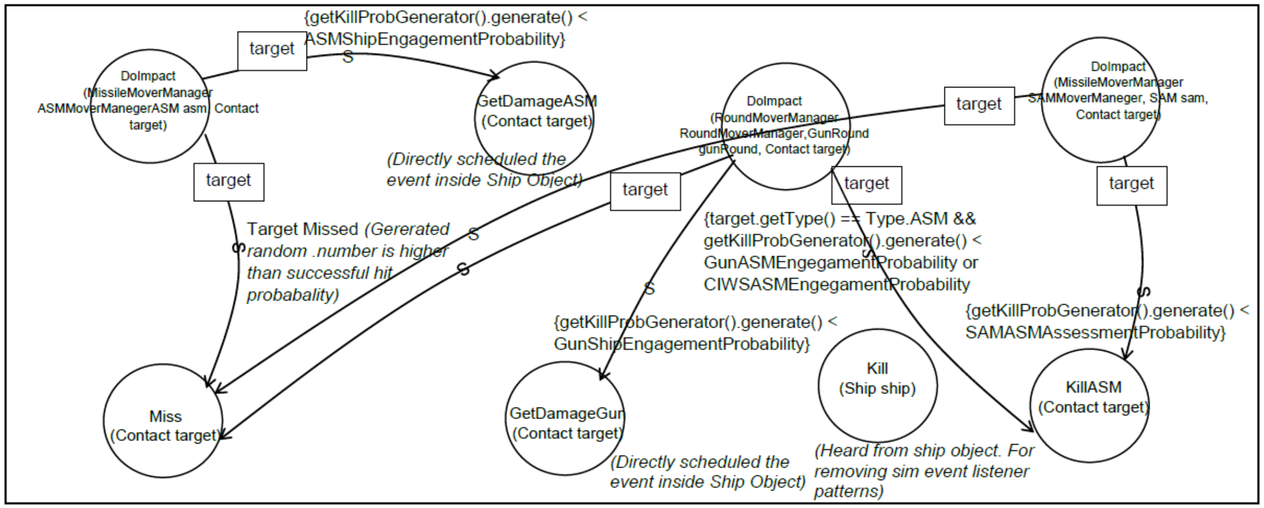


Figure 30. Adjudicator Component Event Graph.

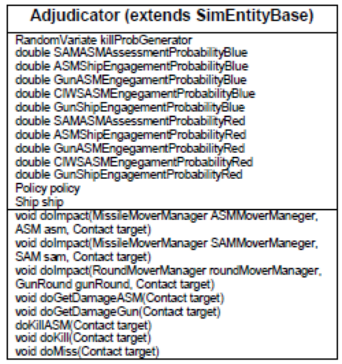


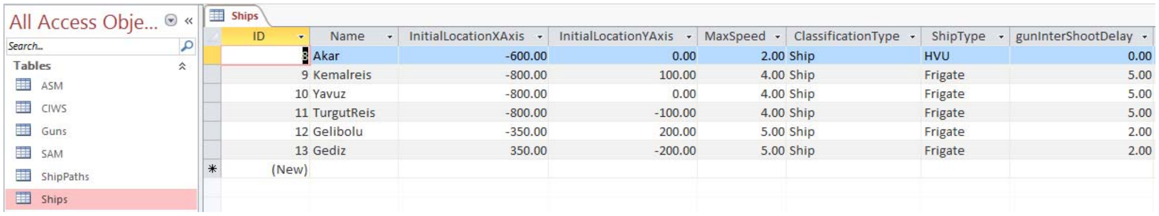
Figure 31. Adjudicator Component UML Diagram.

P. USAGE OF MICROSOFT ACCESS DATABASE, READING DATA FROM DATABASE

Microsoft Access database software can be used to build simple personal database management systems. In this study, we need to make inquiries about various types of systems. For example, for ships, what guided missiles, guns, and radar components do they have? We also need lots of additional information about weapons or systems of ships, such as the ranges of guns, guided missiles, and sensors. That's why it was an inevitable need to include data reading from a database query system. Nevertheless, because the tool that we built does not include very big and complex data queries and data inputs, a simple database management system like Microsoft Access database is sufficient. The application and study is open for necessary changes on database management as it grows.

Since it is almost impossible to include all of the data needed in a single spreadsheet with the required level of complexity, spreadsheets were not used. In this situation, handling the data in multiple spreadsheets would decrease the power of the application. Additionally, the advantages that were gained from using a database are as follows:

- The ability to link table elements. For instance, the weapon data table is linked to the main data table of ships (see Figure 32) and the weapon table's elements can be displayed and selected from the associated column of main table.



ID	Name	InitialLocationXAxis	InitialLocationYAxis	MaxSpeed	ClassificationType	ShipType	gunInterShootDelay
8	Akar	-600.00	0.00	2.00	Ship	HVU	0.00
9	Kemalreis	-800.00	100.00	4.00	Ship	Frigate	5.00
10	Yavuz	-800.00	0.00	4.00	Ship	Frigate	5.00
11	TurgutReis	-800.00	-100.00	4.00	Ship	Frigate	5.00
12	Gelibolu	-350.00	200.00	5.00	Ship	Frigate	2.00
13	Gediz	350.00	-200.00	5.00	Ship	Frigate	2.00
(New)							

Figure 32. Ship Data Table in Microsoft Access Database.

- The ability to read from a single file in a database. Manipulating the data is easier than using multiple spreadsheets.
- Since all the tables are linked, any change in the data table exists in the main database.

The question was how to read from a data table, which is possible in the Java programming language. An access database driver class created by Professor Arnold H. Buss has been used as a main tool. With that class, the ability to create a connection to an Access Database is gained.

IV. ANALYSIS OF THE AAW MODEL

A. INTRODUCTION

After building the simulation, the most important part of the study is using it to obtain insight into the factors affecting AAW. Outputs of the simulation, defined later as MOEs, and factors, such as input parameters, and appropriate design of experiments are crucial for the analysis.

The AAW Analysis Model has a well-defined terminating state that is triggered either by the HVU reaching its goal or the HVU's destruction by enemy ships. That's why, before beginning any analysis, we have to state that it is a terminating simulation and conduct our analysis accordingly. For a terminating simulation with randomness, like the AAW Analysis Model, it is critical to have as many replications as we can, for different parameters, to capture the variability in the output and examine the dependencies between factors and output results.

B. PERFORMANCE MEASURES

The first step in the analysis of the AAW Model is determining the measures of effectiveness (MOEs) to compute or to analyze. The MOEs are those that we can use to analyze the effectiveness of a system (Nakayama, 2008). For the AAW Analysis Model, the HVU's survival is our main interest. As stated previously, the AAW Analysis Model is a terminating simulation that stops for the following two reasons:

- The HVU reaches the goal set at the beginning of the simulation, or
- The HVU's destruction by enemy units.

Because it is a terminating simulation, it has transient MOEs (Nakayama, 2008). The MOEs inside the AAW Analysis Model are as follows:

- The HVU's survival as a binary outcome;
- The HVU's efficiency level (remaining percentage of resilience againist enemy missiles and gun rounds) at the end of the simulation as a continuous output.

Measures of Performance (MOPs) are the factors or input combinations that affect a system's performance. In the AAW Analysis Model, the outcomes are the efficiency of the HVU when the simulation is terminated and the binary outcome of the HVU's survival. So, all the screen ship specifications that affect those outcomes, such as the number of ASMs that a screen ship has, the range of its sensors, and the range of its weapons, are the MOPs in the AAW Analysis Model.

Each iteration for each combination of factors is going to be independent, so that the data collected from that simulation can be analyzed using classical statistics (Sanchez, 2007). Since the outputs are going to be analyzed on a finite time horizon, the initial conditions at the beginning of the simulation may significantly affect the result of the AAW Analysis Model. That's why, assumptions and—for different scenarios—initial conditions are defined separately. Because of time constraints, a limited number of scenarios have been analyzed.

C. SCENARIOS AND ASSUMPTIONS

1. Assumptions

Key AAW Analysis Model assumptions are listed below:

1. Initial conditions may impact the simulation outputs of the AAW Analysis Model. The initial conditions are the main assumptions, and they are stated in scenarios.

2. Probabilities of kill and probabilities of hits for particular types of engagements stated above are assumed to come from a uniform distribution and impact calculations assume they are greater than particular thresholds that are listed for each adjudicator.

   - The ASM-ship engagement probability of hit
   - The SAM-ASM engagement probability of kill
   - The gun-ASM engagement probability of kill
   - The gun-ship engagement probability of hit
   - The CIWS-ASM engagement probability of kill

3. Armstrong (Armstrong, 2005), in his study about stochastic salvo model naval surface combats, proposed a stochastic damage function that is normally distributed with the mean (1/Number of Combat Ships) on the side that is taking an ASM hit, and a standard deviation (1/2.5*Number of Combat Ships). In the AAW Analysis Model, using this approach, a truncated normal distribution for the ASM and the gun damage functions between minimum and maximum damage parameters is assumed. However, parameters of that normal distribution, the minimum and maximum damage parameters vary by the design points. Thus, the parameters of these normal equations are among the input factors of the AAW Analysis Model.

4. Ships are assumed to have 100 efficiency level (remaining percentage of resilience againist enemy ASMs and gun rounds) at the beginning of the scenario for both Blue and Red ships. The efficiency levels of ships decrease with the successful ASM and gun round hits they take. For instance, HVU's efficiency level is a measure of the ship's resilience to hits from ASMs and gun rounds. At time 0, it is 100 and is decreased by a random amount each time the ship is hit.

5. Frigates for both sides are assumed to have surveillance radars for detection; ASMs, Guns, and CIWs as weapons—and specific kinds of engagement radars for ASM, SAM and CIWS engagements.

6. The HVU on the Blue side is assumed to have only CIWS and a surveillance radar.

7. Blue ships and Red ships are not expected to make avoidance maneuvers against each other at the time of detection and they continue on their planned routes.

8. Blue screen ships are assumed to keep the relative positions and distances specified at the beginning of the scenario.

9. Red ships are assumed to keep their speeds and routes according to the patrol paths specified at the beginning of the scenario.

10. Detection rates of surveillance radars for both Blue and Red ships are constant during the scenario and time until detection come from an exponential distribution. Blue and Red ships may have different parameters for detection rates of radars for each design point.

11. Ships that are on the same side do not coordinate their attacks. They attack every live target classified as an enemy contact.

12. Engagements of SAM, gun and CIWS weapons against ASMs are made primarily according to the state and distance of targets on the given order below as part of a layered defense:

- SAMs
- Guns
- CIWS

13. Engagements against ships are made according to distances and states of the enemy ships with the given order below:

- ASMs
- Guns

14. Targets are not prioritized. Targets are engaged by the detection order of the ships.

15. The ASM, the SAM launch, and the gun/CIWS engagement policy by the ships are as follows:

- ASM launch authorization is given if the conditions below occur:
    - Any impact of the previously launched ASM if there is one. (These impacts may occur between the ASM and enemy ships, the ASM-enemy defensive gun and the ASM-enemy defensive SAM.)
    - Any detection of an enemy ship if the enemy ship is classified alive and the enemy ship is not inside the gun range. (An alive classification is primarily made by checking the movement of the enemy ship.)
    - A sufficient number of ASMs are on the ship.
    - The ship’s efficiency level is sufficient.
- The SAM launch may happen any number of times according to the Shoot-Look-Shoot policy or Shoot-Shoot-Look policy. Looking represents checking whether the target is still alive after an intercept should have occurred. The SAM engagement can happen if:
    - The target is detected, classified as an ASM and moving.

        - The target is not inside the CIWs or the gun range.

    - Gun engagement to ASMs may occur if:

        - The target is detected and classified as an ASM and moving.

        - The target could not be destroyed by a SAM and is inside the gun's range.

    - A gun engagement to ships may occur if:

        - A target is detected, classified as enemy ship and moving.

        - The target is inside the gun range.

    - The CIWS engagement to ASMs may occur if:

        - The target is detected, classified as ASM and moving.

        - The target is inside the CIWS range.

16. The probability of hit for each engagement type is constant during each run for each design point.

17. Engagement sensors are cookie-cutter sensors for only engaging detected targets.

18. Enemy targets detected by sensors are not directly passed to the ship for engagement. Each sensor is assumed to have a distortion with a rotated bivariate normal distribution scaled by the target's relative distance and relative angle according to the ship's position. These distortion parameters for sensors may also change in each design point. After the target is detected, the distortion does not change over time.

19. Sensor and weapon positions on the ships and their effects are ignored.

20. Screen ships take their positions at maximum speeds at the beginning of the scenario and maintain them with the HVU, taking its speed and route into consideration, and using the follower mover manager.

21. Real data is not used for modeling weapons and sensors. Instead, by varying parameters, effects on output are examined.

22. Ships are assumed to have a check interval for target positions, states, and velocity updates. This check interval is also introduced as a factor inside the AAW Analysis Model.

23. Any air threat other than ASMs and gun rounds by enemy ships are assumed not to exist inside the scenarios.

24. The only surface threats are frigates of particular types.

25. Any soft kill methods for AAW threats are ignored.

26. Ships do not make any evasive maneuvers for incoming air threats.

27. The minimum distance of impacts for each ship's weapon is also introduced as a factor to analyze.

28. Missiles for both sides are assumed to have the same quality of guidance in the simulation, and they use the same missile mover manager.

29. A ship surveillance sensor range is the same as its ASM range. In other words, a ship can engage any target which it detects with its surveillance sensor.

30. Gun rounds and CIWS rounds do not have guidance.

31. All the moving entities inside the simulation use Basic Linear Movement. Basic Linear Movement is assumed to capture all the effects of movements in those particular types of missions and operational areas.

32. The simulation is conducted in a 2D environment. 3D effects, like the height of radars, ships, and missile engagements, and their effects are ignored.

33. Air propagation conditions and the uncertainty they introduce are assumed to be captured by radar detection rates, distortion factors and mean detection times.

34. Red forces are assumed to have identical frigates for each scenario. That is, for instance, in any scenario all the Red frigates have the same ASM launch delay after classification of a target as an enemy.

35. Blue forces are assumed to have identical frigates. That is, for instance, in any scenario, all the Blue frigates will have the same ASM launch delay after classification of a target as an enemy. Blue ships properties may be different from Red ship properties to capture the effects of the factors for both enemy and friendly units.

36. The classification of enemy units is assumed to be always correct. That is, no misclassifications of targets are introduced in the simulation. In other words, an enemy unit cannot be classified friendly or vice versa. This effect is assumed to be captured by classification times.

2. Scenarios

In each scenario, Red forces are making patrols in their predefined areas of operation and paths. Red's mission is engaging the blue ships detected in their patrol area and destroying them. The policy about engagement priority is not defined. They engage according to the classification of an enemy unit and their primary criterion is detection order.

The threat axis at the initial scenario condition is south and north. As the Blue convoy moves on its path and Red ships make their patrols on their paths, the threat axis is subject to change and may be 360 degrees according to route changes.

a. Scenario 1

In scenario 1, the blue convoy consists of three frigates and the HVU and the Red force consists of two frigates. The Blue screen ships protect the HVU. The mission of the Blue convoy is to reach the predefined convoy goal position by protecting the HVU. The HVU's survival is the crucial objective of the mission. If the HVU is destroyed by the Red ships, the mission fails.

b. Scenario 2

In scenario 2, the Blue convoy consists of four frigates and the HVU and the Red force consists of four frigates. Blue screen ships protect the HVU. The mission of the Blue convoy is to reach the predefined convoy goal position by protecting the HVU. The HVU's survival is the crucial goal of the mission. If the HVU is destroyed by the Red ships, the mission fails.

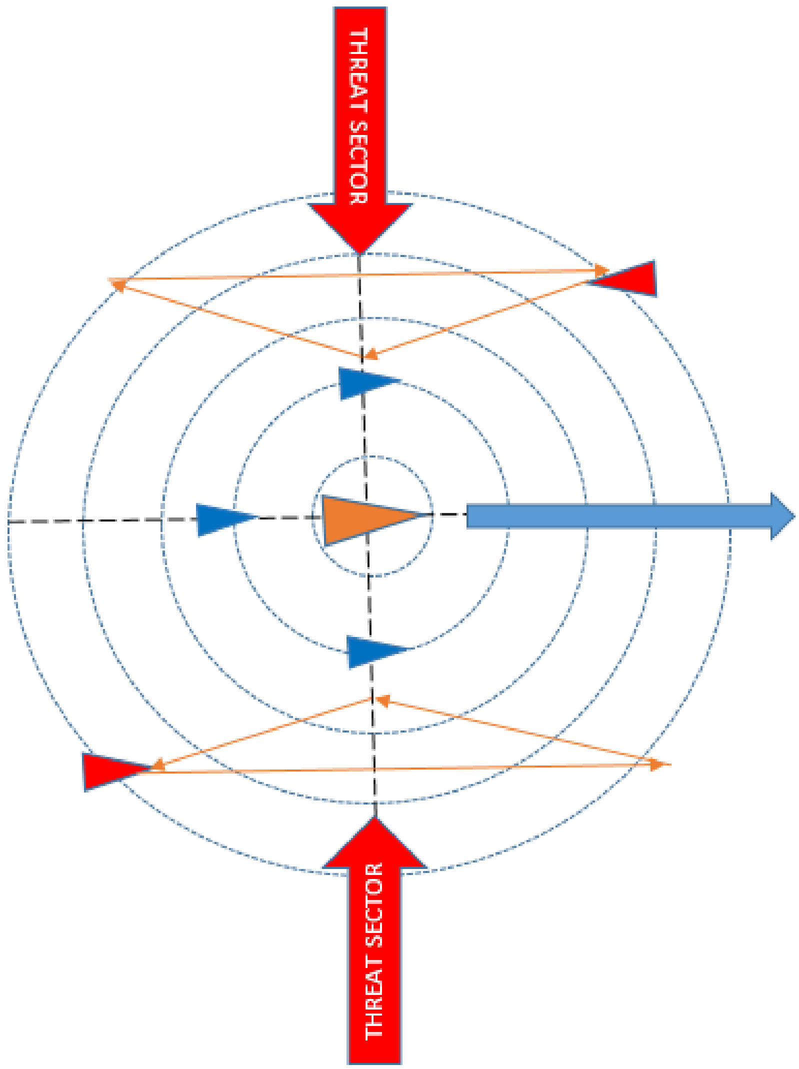


Figure 33. Scenario 1 Initial Conditions.

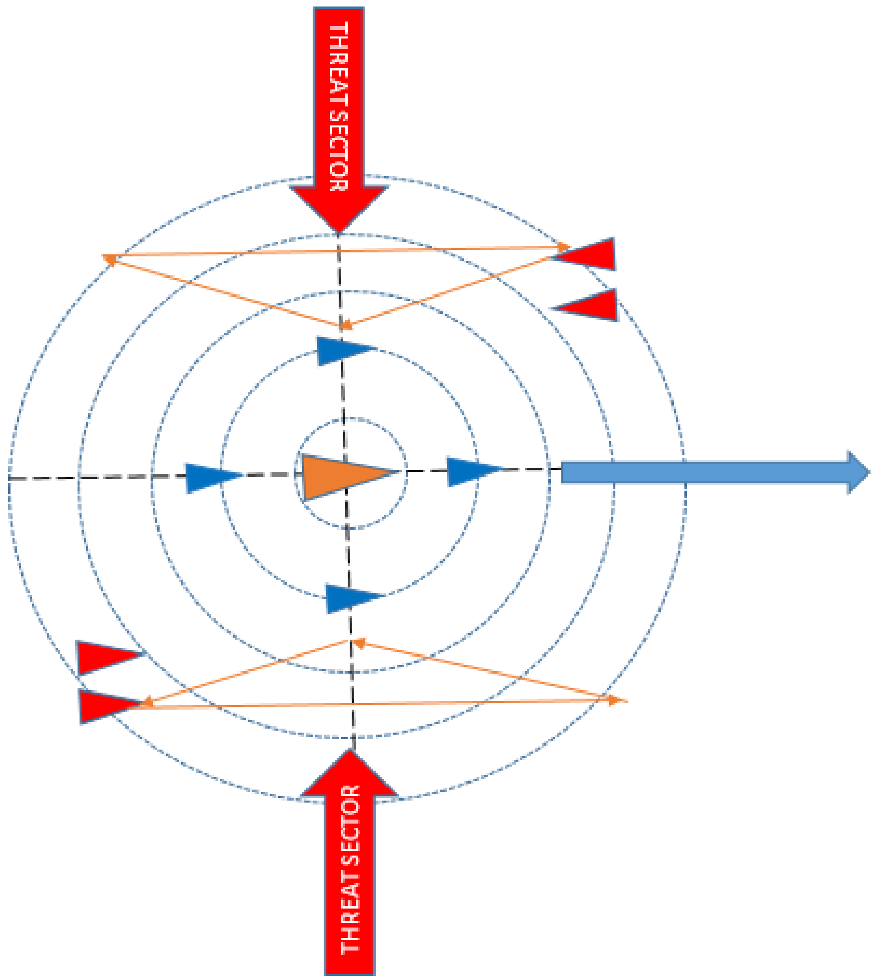


Figure 34. Scenario 2 Initial Conditions.

c. Scenario 3

In scenario 3, the Blue convoy consists of three frigates and the HVU and the Red force consists of four frigates. Blue screen ships protect the HVU. The mission of the Blue convoy is to reach the predefined convoy goal position by protecting the HVU. The HVU's survival is the crucial goal of the mission. If the HVU is destroyed by the Red ships, the mission fails.

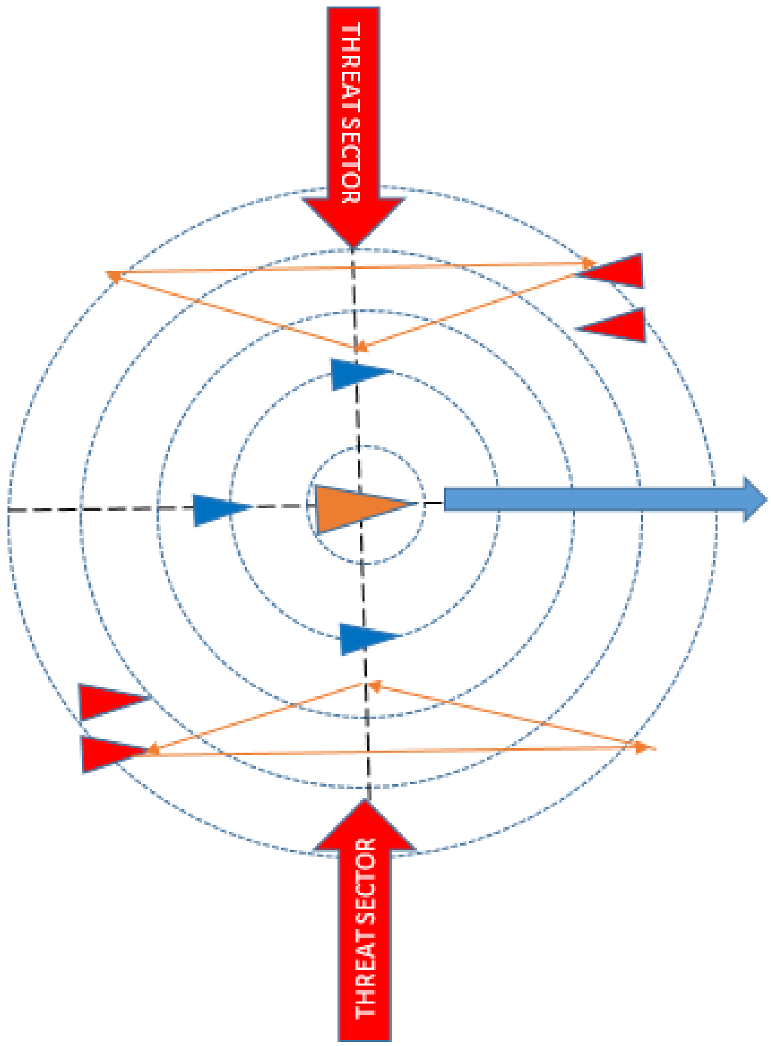


Figure 35. Scenario 3 Initial Conditions.

D. AAW ANALYSIS MODEL FACTORS

Sanchez and Wan (2012) explains potential factors in simulations as follows: “Potential factors in a simulation are the input parameters or the distributional parameters of a simulation model” (p.4). The factors for Blue units inside the AAW Analysis Model correspond to MOPs. However, factors do not necessarily correspond to the input parameters of the simulation. For instance, in the AAW Analysis Model, keeping some attributes of Red ships and changing the attributes of Blue ships over a range may be a more appropriate way to conduct analysis (Sanchez & Wan, 2012).

E. DESIGN OF EXPERIMENTS

The first thing that we have determined for the analysis of the AAW Model is the factors that affected the simulation output, which is the binary outcome of HVU survival when the simulation is terminated. In the DOE, factors are independent input variables that potentially impact on the output. In a simulation experiment, the number of factors we have depends on the model and what we want to analyze. Each of the factors may take a variety of values. Each of the possible values a factor can take is called a level in the DOE. The first goal of the DOE is identifying the factors that are the most impactful on the output or response variable. A second goal is identifying the form of the impact of each factor on the response variable as a function. These can include, but are not limited to, linear or quadratic relationships or even the interactions between factors (Sanchez & Wan, 2012). For instance we want to quantify how the detection rate of the Blue ships affects the efficiency or the survival of the HVU.

The AAW Analysis Model consists of many factors and parameters that may affect the simulation outputs. The HVU's efficiency is our primary MOE, and we are going to investigate effects of the factors previously stated under the assumptions. The AAW Analysis Model is moderately complex and has a moderate level of resolution. To understand the effects of all the factors, an efficient design of experiments is crucial.

An efficient DOE can efficiently explore this dimensionality at a tiny fraction of computational cost relative to a full factorial exploration. If we suppose a simulation with 100 factors and decide to explore all the combinations of those 100 factors using a full factorial design with only two levels, we would require 2^{100} design points. In that case, even with a super computer that has 16 petaflop capacity—assuming each run takes one second—total simulation run time would take millions of years. A couple of design alternatives illustrate the efficient design of experiments. For example, one of them is Resolution V fractional factorial design; in the case of 100 factors with two levels we require 32,768 design points (Sanchez & Wan, 2012). In that case, with an 8-core desktop (costing roughly $1000), we would have finished a set of experiments "takes a more reasonable one minute to run" in 2.85 days (Sanchez & Wan, 2012). Some other design alternatives we can have almost the same efficiency and even achieve better insights for the model.

It is very useful to classify the factors on several types:

- *Quantitative or qualitative*: Quantitative factors take numerical values; on the other hand, qualitative factors do not. However, they may be assigned codes. In the AAW Analysis Model, for instance, the detection rate, the ASM range, or SAM ranges are quantitative factors, whereas, ship policy types are qualitative.

- *Discrete or Continuous (for quantitative factors only)*: Discrete factors are the ones that have levels only at certain levels. However, continuous factors may have any real value in a specified interval. The total numbers of SAMs or ASMs in the AAW Analysis Model are examples of discrete factors.

- *Binary or Not:* Binary factors are naturally bounded to two levels, like classification. In the AAW Analysis Model, the HVU's survival is binary. However, it is a response variable. Inside the AAW Analysis Model, we do not have any binary factors.

- *Controllable or uncontrollable*: Even though in a simulation experiment all factors can be manipulated, in real life some factors cannot be controlled by operators. For instance, in the AAW Analysis Model, enemy ship attributes are uncontrollable factors in real life. Nevertheless, we will leverage the advantage of the simulation to make a robust analysis. We will assume we can also change enemy ship's attributes. Thus, they will also be treated as simulation factors (Sanchez & Wan, 2012).

The factors to be analyzed inside the AAW Analysis Model are listed as follows:

1. Discrete Factors

a. Controllable

- Blue ship SAM launch policy (Shoot-Look-Shoot/Shoot-Shoot-Look)
- Blue ship number of SAMs to launch between each look
- Blue ship total number of ASMs in each ship
- Blue ship total number of SAMs in each ship

b. Uncontrollable

- Red ship SAM launch policy (Shoot-Look-Shoot/Shoot-Shoot-Look)
- Red ship number of SAMs to launch between each look
- Red ship total number of ASMs in each ship
- Red ship total number of SAMs in each ship

2. Continuous Factors

a. Controllable

- Blue Ship inter-shot delays between gun fires
- Blue Ship inter-shot delays between CIWS fires
- Blue ship classification time
- Blue ship detection rate for surveillance sensor
- Blue ship ASM launch delay
- Blue ship SAM launch delay
- Blue ship ASM range
- Blue ship SAM range
- Blue ship gun range
- Blue ship CIWS range
- Blue ship ASM red ship damage probability

- Blue ship gun red ship damage probability
- Blue ship gun round red ASM kill probability
- Blue ship SAM Red ASM kill probability
- Blue ship CIWS rounds red ASM kill probability
- Close enough distance (CED) for impact of blue ASM
- CED for impact of blue SAM
- CED for impact of blue gun round
- CED for impact of blue CIWS round
- Blue ship surveillance sensor distortion factors
- Blue ship engagement sensor distortion factors
- Blue ship ASM and gun minimum and maximum damages
- Blue ship maximum speed
- Blue ship weapon speeds
- Blue ship target check interval
- Blue ship surveillance sensors time to detection
- Blue ship efficiency threshold
- HVU maximum speed
- HVU CIWS range
- HVU CIWS CED for red ASM impact
- HVU CIWS inter shoot delay time
- HVU efficiency threshold
- Blue ship ASM damage function parameters
- Blue ship gun damage function parameters

b. Uncontrollable

- Red Ship inter-shot delays between gun fires
- Red Ship inter-shot delays between CIWS fires

- Red ship classification time
- Red ship detection rate for surveillance sensor
- Red ship ASM launch delay
- Red ship SAM launch delay
- Red ship ASM range
- Red ship SAM range
- Red ship gun range
- Red ship CIWS range
- Red ship ASM red ship damage probability
- Red ship gun red ship damage probability
- Red ship gun round red ASM kill probability
- Red ship SAM Red ASM kill probability
- Red ship CIWS rounds red ASM kill probability
- CED for impact of blue ASM
- CED for impact of blue SAM
- CED for impact of blue gun round
- CED for impact of blue CIWS round
- Red ship surveillance sensor distortion factors
- Red ship engagement sensor distortion factors
- Red ship ASM and gun minimum and maximum damages
- Red ship maximum speed
- Red ship weapon speeds
- Red ship target check interval
- Red ship surveillance sensors time to detection
- Red ship efficiency threshold
- Red ship ASM damage function parameters

- Red ship gun damage function parameters

Mathematically, if we denote $X_1, X_2, X_3, ...X_n$ as *n* factors in our simulation experiment and let Y be the response variable, which is the HVU's survival or efficiency in our model. What we are interested in is building a response metamodel that approximates and represents relationships between factors and response variables in our simulation. To reach that goal, we can use statistical methods such as regression models (Sanchez & Wan, 2012). If we can create a good metamodel of AAW analysis simulation and have reasonable insights that we can prove some concepts in AAW, this model can also be expanded and used for further purposes. It can also help us explore some concepts in AAW that are not explored and gain better understanding of how AAW ship design should be, and what kind of ships should be sent to naval convoy operations and what tactics they should use.

F. EXPLORATION OF POTENTIAL EXPERIMENTAL DESIGNS

Many design alternatives are available in the literature, and it is almost impossible to analyze all of the possible alternatives in experimental designs. The most useful and suitable ones for the AAW Analysis Model are a small subset of them. The first and most straightforward approach for experimental designs are gridded or factorial designs (Sanchez & Wan, 2012). According to Sanchez and Wan, among gridded or factorial designs, if we are looking for inspection of only linear effects and interactions, coarse grids (2^k factorials) are the best and the most efficient design. On the other hand, fine grids (more than two levels for factors) may provide more detailed information about the response—such as a nonlinear relationship. That method provides efficiency for building metamodels of response variables. Resolution five fractional factorial designs (RVFF) allow linear main effects and interactions to be explored. They look like the best choice if we have few levels of quantitative factors or qualitative factors with two levels. By expanding RVFFs to central composite designs, we may gain more information about non-linear effects.

If the number of factors is large, as it is in the AAW Analysis Model, more efficient design alternatives are required. Latin hypercube (LH) designs have been proved to be good for exploring complex simulations when we have little information about response surfaces (Sanchez & Wan, 2012).

We can flexibly construct efficient designs by using LH designs. Additionally, we can have more efficient space-filling than we have in factorial designs with orders of magnitude less sampling and enough information about the center of experimental design.

The most useful and straightforward design possibilities for the AAW Analysis Model can be listed as follows (Sanchez & Wan, 2012):

- 2^k Factorial Designs (Coarse Grids)
- m^k Factorial Designs (Finer Grids)
- 2^{k-p} Resolution 5 Fractional Factorial and Central Composite Designs
- Space Filling Designs, such as Latin hypercubes (LHs)

As previously stated, factorial (or gridded) designs may look like the most straightforward way to design the experiments. Nevertheless, they increase the design points required by many orders of magnitude over other approaches. For the AAW Analysis Model, in the simplest case, we would have 99 factors, most of which are continuous. Even if we were able to consider them with only two levels, we would have 2^{99} factors, which would require trillions of years of running to obtain all these design point runs, even with supercomputers like roadrunner.

Cioppa and Lucas (Cioppa & Lucas, 2007) came up with nearly orthogonal Latin hypercube (NOLH) designs that provides the excellent space-filling for a small or moderate number of factors up to 29. Hernandez, Lucas, and Carlyle, 2012, expanded on the number of factors that can be investigated with NOLHs. This provides an ability to examine much denser designs with much less effort compared to full factorial and fractional factorial designs. For instance, with 257 design points we can efficiently construct a designed set of experiments with

29 factors, whereas, we would need 2^{29} design points with a full factorial design even if we only consider each factor with 2 levels. Coming up with only two levels for the continuous factors of the AAW Analysis Model is almost impossible. And doing so would make it impossible to identify response thresholds or non-linear effects. Additionally, we would obtain less space-filling. Assuming each design point takes one second to run, with the NOLH, it would take under five minutes to run a single replication of each design point on one processor. However, for a 29 factorial design, this would take 17 years to run under the same conditions (Sanchez & Wan, 2012).

The AAW Analysis Model includes 99 factors with 8 discrete and 91 continuous ones that we wish to explore. Thus, a denser DOE experiment is required. Vieira in 2012 (Vieira, Sanchez , Kienitz, & Belldarrain, 2012) came up with a Nearly Orthogonal Nearly Balanced (NOB) design which allows analyzing 10 blocks of 20 *k*-level factors (*k*=2,3,…11) and 100 continuous factors with 512 design points (Vieira et al., 2012). This design has a maximum absolute pairwise correlation of 3.56%, which is an acceptable level for regression analysis. In statistics, having all the factors orthogonal—in other words, having a correlation of zero among the factors—is the most favorable situation to come up with the best metamodel. However, for this many factors, where orthogonality is impossible, the NOB is a good solution. So, the AAW Analysis Model DOE is determined to be built using the NOB design.

More replications for each design point are highly favorable to capture variability inside the model. So, the AAW Analysis Model is run 1000 times for each design point with different scenarios. This ensures that a standard error on the probability of HVU survival is no greater than .016. The ranges that are defined for each factor to create a NOB design using NOB_Mixed_512DP_v1 (Vieira et al., 2012) are shown in Table 1.

Table 1. Controllable Factor Ranges and Explanations (for Each Blue Ship).

#	Factor	Range	Unit	Explanation
1	Blue ship policy	1–2	-	“Shoot-Look-Shoot”(1) or “Shoot-Shoot-Look”(2) Policies
2	Blue ship number of times to shoot before look	1–3	-	The number of times that SAM is launched before each look if the “ShootShootLook” policy is on force
3	Blue ship total number of ASMs	8–18	-	Total number of ASMs
4	Blue ship total number SAMs	16–26	-	Total Number of SAMs
5	Blue ship inter-shot delays between gun fires	5–25	seconds	The time delay between gun shots
6	Blue ship inter-shot delays between CIWS fires	1–5	seconds	The time delay between CIWS fires
7	Blue ship classification times	0–5	minutes	The deterministic time that is needed to classify an enemy unit
8	Blue ship detection rate for surveillance sensors	1–10	# targets classified / minutes	The surveillance sensor detection time comes from exponential distribution with the parameter of the detection rate.
9	Blue ship ASM launch delay	5–45	seconds	The time delay that ASM is launched after classification of a target as an enemy
10	Blue ship SAM launch delay	5–25	seconds	The time delay to launch a SAM after classification of a target as an

#	Factor	Range	Unit	Explanation
				ASM
11	Blue ship ASM range	100–250	NM	The range of ASMs (Same with the range of surveillance sensor)
12	Blue ship SAM range	75–150	NM	The range of SAMs (Same with the range of SAM engagement sensor)
13	Blue ship gun range	50–100	NM	Range of gun (Same with the range of gun engagement sensor)
14	Blue ship CIWS range	25–75	NM	Range of CIWS (Same with the range of CIWS engagement sensor)
15	Blue ship ASM Red ship damage probability	0–1	-	Probability that one Blue ASM can damage Red ship provided that it has impacted
16	Blue ship gun Red ship damage probability	0–1	-	Probability that one Blue gun round can damage Red ship provided that it has impacted
17	Blue ship gun round Red ASM kill probability	0–1	-	Probability that one Blue gun round can eliminate one Red ASM provided that it has impacted
18	Blue ship SAMs Red ASM kill probability	0–1	-	Probability that one Blue SAM can eliminate one Red ASM provided that it has impacted

#	Factor	Range	Unit	Explanation
19	Blue ship CIWS round Red ASM kill probability	0–1	-	Probability that one Blue CIWS round can eliminate one red ASM provided that it has impacted
20	CED for impact of Blue ASMs	1–10	meters	The distance required for an ASM to impact
21	CED for impact of Blue SAM	1–10	meters	The distance required for a SAM to impact
22	CED for impact of Blue gun round	1–10	meters	The distance required for a gun round to impact
23	CED for impact of Blue CIWS rounds	1–10	meters	The distance required for a CIWS round to impact
24	Blue ship surveillance sensor distortion mean on x axis	1–10	meters	Mean of rotated bivariate normal distribution on x axis for surveillance sensors
25	Blue ship surveillance sensor distortion mean on y axis	1–10	meters	Mean of rotated bivariate normal distribution on y axis for surveillance sensors
26	Blue ship surveillance sensor distortion standard deviation on x axis	1–10	centimeters	Standard deviation of rotated bivariate normal distribution on x axis for surveillance sensors
27	Blue ship surveillance sensor distortion standard deviation on y axis	1–10	centimeters	Standard deviation of rotated bivariate normal distribution on y axis for

#	Factor	Range	Unit	Explanation
				surveillance sensors
28	Blue ship engagement sensor distortion mean x axis	1–10	meters	Mean of rotated bivariate normal distribution on x axis for engagement sensors
29	Blue ship engagement sensor distortion mean y axis	1–10	meters	Mean of rotated bivariate normal distribution on y axis for engagement sensors
30	Blue ship engagement sensor distortion standard deviation on x axis	1–10	centimeters	Standard deviation of rotated bivariate normal distribution on x axis for engagement sensors
31	Blue ships engagement sensors distortion standard deviation on y axis	1–10	centimeters	Standard deviation of rotated bivariate normal distribution on y axis for engagement sensors
32	Blue ship ASM minimum damage	10–25	%Efficiency	Minimum damage that ASM can give
33	Blue ship ASM maximum damage	40–100	%Efficiency	Maximum damage that ASM can give
34	Blue ship gun minimum damage	5–10	%Efficiency	Minimum damage that gun round can give
35	Blue ships gun maximum damage	20–40	%Efficiency	Maximum damage that gun round can give
36	Blue ASM maximum speed	100–650	Knots	Maximum speed of ASM
37	Blue SAM maximum speed	800–2500	Knots	Maximum speed of SAM

#	Factor	Range	Unit	Explanation
38	Blue gun round maximum Speed	700–1700	Knots	Maximum speed of gun round
39	Blue CIWS round maximum speed	800–2500	Knots	Maximum Speed of CIWS round
40	Blue ship surveillance sensor time delay until detection	1–10	seconds	Required time to detect a contact for surveillance sensor
41	Blue ship efficiency threshold	10–50	Out of 100	Threshold that is required to destroy a blue screen ship.
42	HVU maximum speed	2–10	Knots	Maximum speed of HVU
43	HVU CIWS Range	25–75	NM	HVU CIWS range
44	HVU CIWS close enough distance for red ASM impact	1–10	meters	The distance required for an HVU CIWS round to impact
45	HVU CIWS inter-shot delay times	1–5	seconds	The delay time between CIWS fires
46	HVU efficiency threshold	10–50	Out of 100	Threshold that is required to destroy an HVU.
47	Blue ship ASM damage mean	25–50	%Efficiency	The mean parameter of ASM damage function, which comes from a normal distribution
48	Blue ship ASM damage standard deviation	10–20	%Efficiency	The standard deviation parameter of ASM Damage function that comes from a normal distribution
49	Blue ship gun damage mean	10–20	%Efficiency	The mean parameter of gun damage function that comes from a normal distribution

#	Factor	Range	Unit	Explanation
50	Blue ship gun damage standard deviation	4–8	%Efficiency	The standard deviation parameter of gun damage function that comes from a normal distribution
51	Blue ship maximum speed	11–20	Knots	Maximum Speed of Blue Ships
52	Blue ship target check interval	5–20	Seconds	How frequent the enemy target is checked after detection (No new detection is made. The target has the same distortion)

The uncontrollable factors, their ranges, and explanations are shown in Table 2.

Table 2. Uncontrollable Factor Ranges and Explanations (for Each Red Ship).

#	Factor	Range	Unit	Explanation
1	Red ship policy	1–2	-	“ShootLookShoot” (1) or “ShootShootLook” (2) Policies
2	Red ship number of times to shoot before look	1–3	-	The number of times that SAM is launched before each look if “ShootShootLook” policy is on force
3	Red ship total number of ASMs	8–18	-	Total number of ASMs
4	Red ship total number SAMs	16–26	-	Total Number of SAMs
5	Red ship inter-shot delays between gun Fires	5–25	seconds	The time delay between gun shots
6	Red ship inter-shot delays between CIWS fires	1–5	seconds	The time delay between CIWS

#	Factor	Range	Unit	Explanation
				fires
7	Red ship classification times	0–5	minutes	The deterministic time that is needed to classify an enemy unit
8	Red ship detection rate for surveillance sensors	1–10	# targets classified / minutes	Surveillance sensor detection time comes from exponential distribution with the parameter of detection rate.
9	Red ship ASM launch delay	5–45	seconds	The time delay that ASM is launched after classification of a target as an enemy
10	Red ship SAM launch delay	5–25	seconds	The time delay to launch a SAM after classification of a target as an ASM
11	Red ship ASM range	100–250	NM	The range of ASMs (Same with the range of surveillance sensor)
12	Red ship SAM range	75–150	NM	The range of SAMs (Same with the range of SAM engagement sensor)
13	Red ship gun range	50–100	NM	Range of gun (Same with the range of gun engagement Sensor)
14	Red ship CIWS range	25–75	NM	Range of CIWS (Same with the range of CIWS engagement sensor)

#	Factor	Range	Unit	Explanation
15	Red ship ASM Blue ship damage probability	0–1	-	Probability that one Blue ASM can damage Red ship provided that it has impacted
16	Red ship gun Blue ship damage probability	0–1	-	Probability that one Blue gun round can damage Red ship provided that it has impacted
17	Red ship gun round Blue ASM kill probability	0–1	-	Probability that one Blue gun round can eliminate one Red ASM provided that it has impacted
18	Red ship SAMs Blue ASM kill probability	0–1	-	Probability that one Blue SAM can eliminate one red ASM provided that it has impacted
19	Red ship CIWS round Blue ASM kill probability	0–1	-	Probability that one Blue CIWS round can eliminate one Red ASM provided that it has impacted
20	CED for impact of Red ASMs	1–10	meters	The distance required for an ASM to impact
21	CED for impact of Red SAM	1–10	meters	The distance required for a SAM to impact
22	CED for impact of Red gun round	1–10	meters	The distance required for a gun round to impact
23	CED for impact of Red CIWS rounds	1–10	meters	The distance required for a CIWS round to impact
24	Red ship surveillance sensor distortion mean on	1–10	meters	Mean of rotated bivariate normal

#	Factor	Range	Unit	Explanation
	x axis			distribution on x axis for surveillance sensors
25	Red ship surveillance sensor distortion mean on y axis	1–10	meters	Mean of rotated bivariate normal distribution on y axis for surveillance sensors
26	Red ship surveillance sensor distortion standard deviation on x axis	1–10	centimeters	Standard deviation of rotated bivariate normal distribution on x axis for surveillance sensors
27	Red ship surveillance sensor distortion standard deviation on y axis	1–10	centimeters	Standard deviation of rotated bivariate normal distribution on y axis for surveillance sensors
28	Red ship engagement sensor distortion mean x axis	1–10	meters	Mean of rotated bivariate normal distribution on x axis for engagement sensors
29	Red ship engagement sensor distortion mean y axis	1–10	meters	Mean of rotated bivariate normal distribution on y axis for engagement sensors
30	Red ship engagement sensor distortion standard deviation on x axis	1–10	centimeters	Standard deviation of rotated bivariate normal distribution on x axis for engagement sensors
31	Red ships engagement sensors distortion standard deviation on y	1–10	centimeters	Standard deviation of rotated bivariate normal distribution

#	Factor	Range	Unit	Explanation
	axis			on y axis for engagement sensors
32	Red ship ASM minimum damage	10–25	%Efficiency	Minimum damage that ASM can give
33	Red ship ASM maximum damage	40–100	%Efficiency	Maximum damage that ASM can give
34	Red ship gun minimum damage	5–10	%Efficiency	Minimum damage that gun round can give
35	Red ships gun maximum damage	20–40	%Efficiency	Maximum damage that gun round can give
36	Red ASM maximum speed	100–650	Knots	Maximum speed of ASM
37	Red SAM maximum speed	800–2500	Knots	Maximum speed of SAM
38	Red gun round maximum speed	700–1700	Knots	Maximum speed of gun round
39	Red CIWS round maximum speed	800–2500	Knots	Maximum speed of CIWS round
40	Red ship surveillance sensor time delay until detection	1–10	seconds	Required time to detect a contact for surveillance sensor
41	Red ship efficiency threshold	10–50	Out of 100	Threshold that is required to destroy a Blue screen ship.
42	Red ship ASM damage mean	25–50	%Efficiency	The mean parameter of ASM damage function that comes from a normal distribution
43	Red ship ASM Damage standard deviation	10–20	%Efficiency	The standard deviation parameter of ASM damage function that comes from a normal distribution
44	Red ship gun damage mean	10–20	%Efficiency	The mean parameter of gun

#	Factor	Range	Unit	Explanation
				damage function that comes from a normal distribution
45	Red ship gun damage standard deviation	4–8	%Efficiency	The standard deviation parameter of gun damage function that comes from a normal distribution
46	Red ship maximum speed	11–20	Knots	Maximum speed of blue ships
47	Red ship target check interval	5–20	Seconds	How frequent the enemy target is checked after detection (No new detection is made. The target has the same distortion)

G. ANALYSIS OF RESULTS

1. Overview

JMP Pro 12.0 (SAS Institute Inc., 1989) software is used as a statistical analysis tool. Three scenarios were analyzed. These three scenarios were each run 1000 times at each of 512 design points that were created systematically using a NOB Design (Vieira et al., 2012). This resulted in 512,000 simulated AAW battles in each of the three scenarios. Each run took an average of three milliseconds on a personal computer with an Intel (R) Core (TM) i7–4810MQ 2.8 Ghz CPU and 8 GB RAM. Run times varied depending on the number of ships instantiated. This was an expected result, because the number of calculations and objects instantiated increased as the number of ships in the scenario increased. The first MOE was the efficiency level of the HVU and the second MOE was the binary value showing whether the HVU survived or not, which meant the HVU's efficiency level was greater than its efficiency threshold, as previously mentioned. For each scenario, averages of these two MOEs are shown in Table 3.

Table 3. Summary Statistics for Three Scenarios.

Scenario	Number of Blue Ships	Number of Red Ships	Mean HVU Efficiency	HVU's Survival Rate (% replications in which HVU survived)
1	3	2	88.99	94.7
2	4	4	79.97	86
3	3	4	69.66	77.1

In this section a very coarse analysis is made, since it aggregates across the factor settings. The results clearly show that as the number of Blue ships increases relative to Red, the rate of HVU's survival (binary outcome of HVU survival) and the mean efficiency level increases. This result can be seen if scenario 2 and scenario 3 are compared. Boxplots for mean HVU efficiencies and survivals are displayed in Figures 36 and 37.

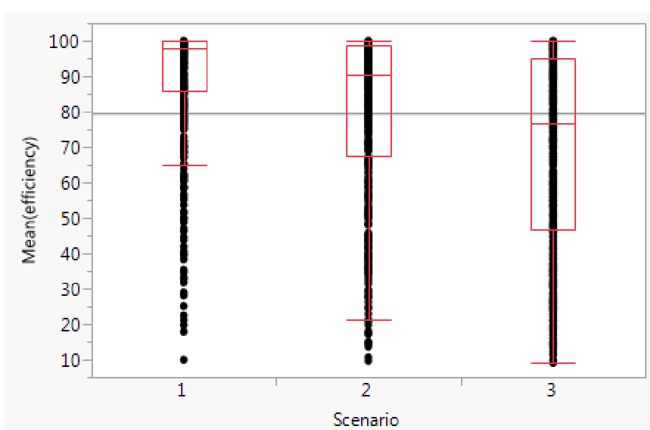


Figure 36. Comparative Boxplots of Mean HVU Efficiency by Scenario.

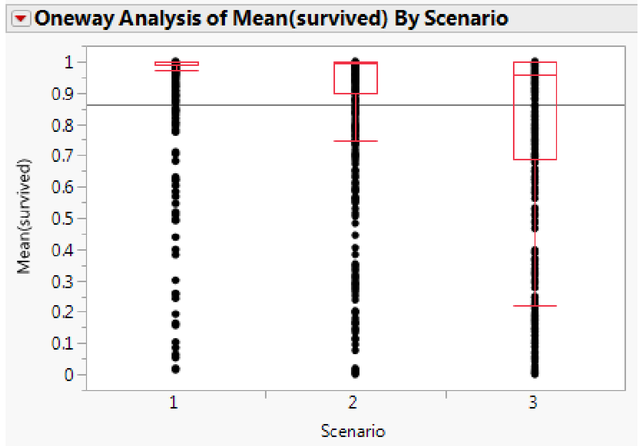


Figure 37. Comparative Boxplots of Mean HVU Survival by Scenario.

Additionally, as the number of Red ships increases, the mean of the HVU's survival and the mean efficiency level of HVU decreases. This can be realized, if we check scenarios 1 and 3. Figures 38, 39 and 40 show the distribution of mean HVU efficiencies and survivals for each scenario depending on each design point.

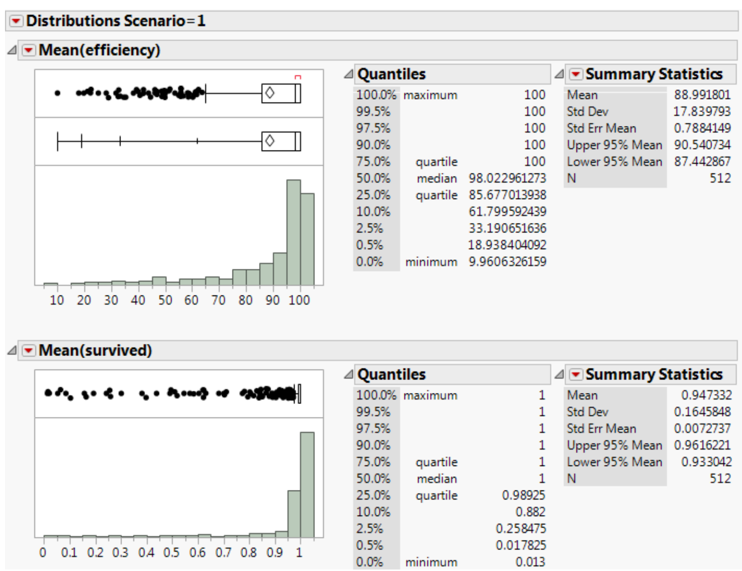


Figure 38. Distribution of Mean Efficiencies and Survivals of HVU in Scenario 1.

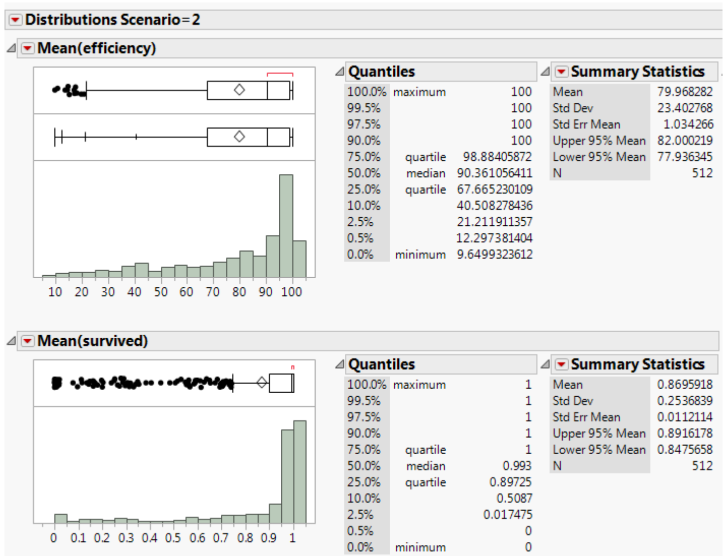


Figure 39. Distribution of Mean Efficiencies and Survivals of HVU in Scenario 2.

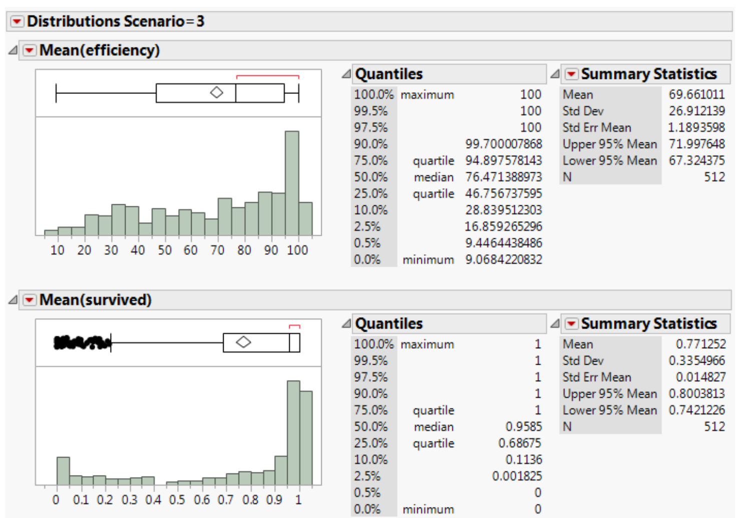


Figure 40. Distribution of Mean Efficiencies and Survivals of HVU in Scenario 3.

The distribution of mean HVU efficiencies and survivals show that they are not normally distributed. Normal Quantile plots for mean HVU efficiencies and survivals also support the skewed distribution.

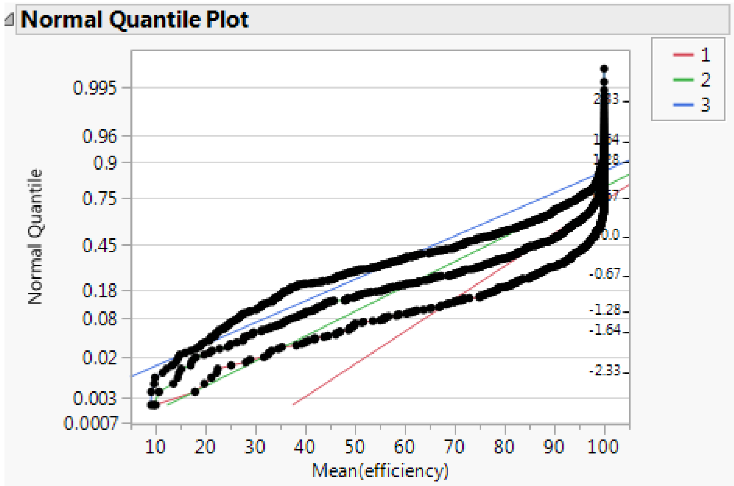


Figure 41. Normal Quantile Plot for Mean HVU Efficiencies.

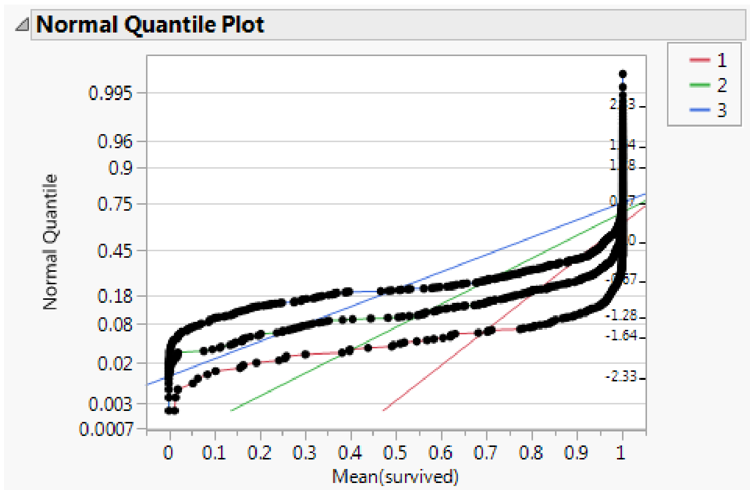


Figure 42. Normal Quantile Plot for Mean HVU Survivals.

Because they are not normally distributed, we used non-parametric comparisons for each pair using the Wilcoxon method (Wackerly, Mendenhall, & Scheaffer, 2002) to compare the means of HVU efficiencies and survivals for each scenario.

Nonparametric Comparisons For Each Pair Using Wilcoxon Method

q*	Alpha
1.95996	0.05

Level	- Level	Score Mean Difference	Std Err Dif	Z	p-Value	Hodges-Lehmann	Lower CL	Upper CL
3	2	-120.232	18.47948	-6.5063	<.0001*	-7.2843	-10.2768	-4.3813
2	1	-151.135	18.42030	-8.2048	<.0001*	-3.4442	-5.3384	-2.0434
3	1	-250.533	18.43961	-13.5867	<.0001*	-14.3538	-17.5934	-11.1678

Figure 43. Non-parametric Comparisons for Mean Efficiencies of HVU at Each Scenario.

Nonparametric Comparisons For Each Pair Using Wilcoxon Method

q*	Alpha
1.95996	0.05

Level	- Level	Score Mean Difference	Std Err Dif	Z	p-Value	Hodges-Lehmann	Lower CL	Upper CL
3	2	-98.109	18.21739	-5.3855	<.0001*	-0.005000	-0.013000	-0.001000
2	1	-147.934	17.44703	-8.4790	<.0001*	-0.002000	-0.003000	0.000000
3	1	-231.676	17.75531	-13.0483	<.0001*	-0.024000	-0.038000	-0.012000

Figure 44. Non-Parametric Comparisons for Mean Survivals of HVU at Each Scenario.

The results show a significant difference among all pairs of means for efficiencies. Also, we see in the non-parametric comparisons for each pair that scenario 2 mean efficiencies, where we have four Blue and four Red ships, is greater than scenario 3 mean HVU efficiencies, with three Blue and four Red ships at an $\alpha = 0.05$ significance level. Additionally, the difference in mean HVU efficiencies between scenario 1 and scenario 3 shows the effect of increasing the number of Red ships. Thus, having more blue ships increases the mean efficiency level of the HVU; whereas, if there are more enemy ships, it decreases the likelihood of survival. Also, the non-parametric comparison tests on mean survival of HVU complement the results, as expected.

2. Analysis of Scenarios

In this section, the data from all three scenarios have been aggregated. Then, partition tree analysis has been made on the aggregate data where we had 1,536,000 simulated AAW battles on the Blue ship properties for a robust analysis. Afterwards, a nominal logistic regression analysis was fit on the binary outcome of the HVU's survival depending on the parameters of Blue ships. Two different analysis methods were adopted to check whether they complement each other.

After analyzing the Blue ship properties, both partition tree and logistic regression methodologies are leveraged to analyze the effects of all the factors, including the Red ship parameters. This analysis led us to understand which properties were the most effective on the survival of the HVU, including the Red ship properties.

There was another option of fitting a least squares regression on the response of mean efficiency levels/survivals on Blue ship parameters up to two-way interactions. This was going to be done by collapsing the efficiency levels/survivals on each design point by taking an average of 1000 runs. However, that violated normality assumption residuals and the constant variance assumption of residual by predicted plots. Also, as aforementioned, response variables were not normally distributed. That's why fitting a multiple linear regression was not a good option for analysis of individual scenarios. The assumption violations for the aggregate data from all three scenarios are seen in Figure 45. In this analysis, mean efficiency of HVU is the response variable and Blue ship parameters are explanatory variables.

Figure 45 demonstrates that it is impossible to get a good fit by fitting a least squares regression for that data. As seen in the figure, the residuals do not come from a normal distribution and show a clear constant variance violation. We can transform the response variable. However, instead of that option, using more appropriate analysis methods for that analysis would be better. So a nominal logistic fit, which assumes the predictors come from a binomial distribution, and a partition tree, which does not require any assumptions about the data, have been adopted. These two methodologies do not particularly make any assumptions about linearity, normality, and homoscedasticity.

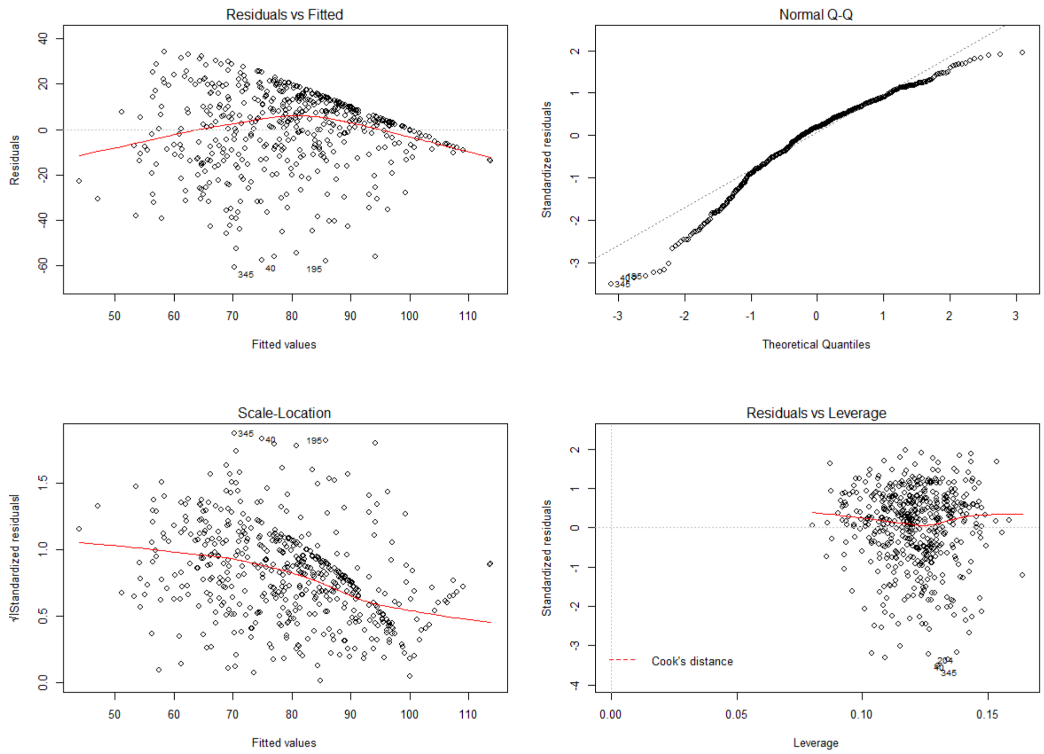


Figure 45. Assumption Violations of Least Squares Regression of Mean Efficiency of HVU.

3. Analysis of All Factors Including Enemy Ship Factors

For some operations, factors of enemy units may have substantial effect on success. If we have that information before starting the operations, we can try to eliminate or decrease the efficiency of the most important factors that enemy units have.

For that purpose, we analyzed all aggregate data from the three scenarios including both Blue ship and Red ship properties, even if they are uncontrollable factors. The same analysis methodology is used to have the most effective factors on the HVU's survival as a binary outcome. First, we analyzed data with a partition tree to classify the binary outcome of the HVU's survival. Afterwards, we fit a nominal logistic regression on all factors.

a. Partition Tree

In this section, we partitioned the aggregate data from all three scenarios using all the factors for Blue ships and Red ships. The simulation had 99 factors, additionally; the number of blue screen ships and the number of Red ships are predictors. Figure 46 is the partition tree that was obtained after 15 splits.

Using partition tree methodology, we try to split the whole data into branches to classify whether the ship is survives or not. This recursive partitioning process creates a decision tree to classify the members of a particular population by splitting into sub populations based on input variables. For our case, the input variables are the Blue ship parameters, and we will try to classify the binary outcome of the HVU's survival.

For a least squares regression, R-squared is the number that indicates how well the model is fits. For our partitioning tree, we try to classify a binary outcome. Receiver Operating Characteristics (ROC) is a plot that demonstrates the performance of a binary classification system by varying its discrimination threshold. So, for partition tree analysis and the nominal logistic regression model, the ROC curve is going to show how well we predict the outcome based on the model. The curve is obtained by calculating true positive rates and false positive rates at varying thresholds and plotting them. To assess the goodness of the ROC curve, we compared the Area Under Curve (AUC). The greater AUC is, the better our model is. During the analysis, AUC over 0.75 is the threshold of goodness.

While splitting the data, the ROC curve was continuously checked. The tree split in Figure 46 shows the split points depending on the response variables of Blue ship parameters.

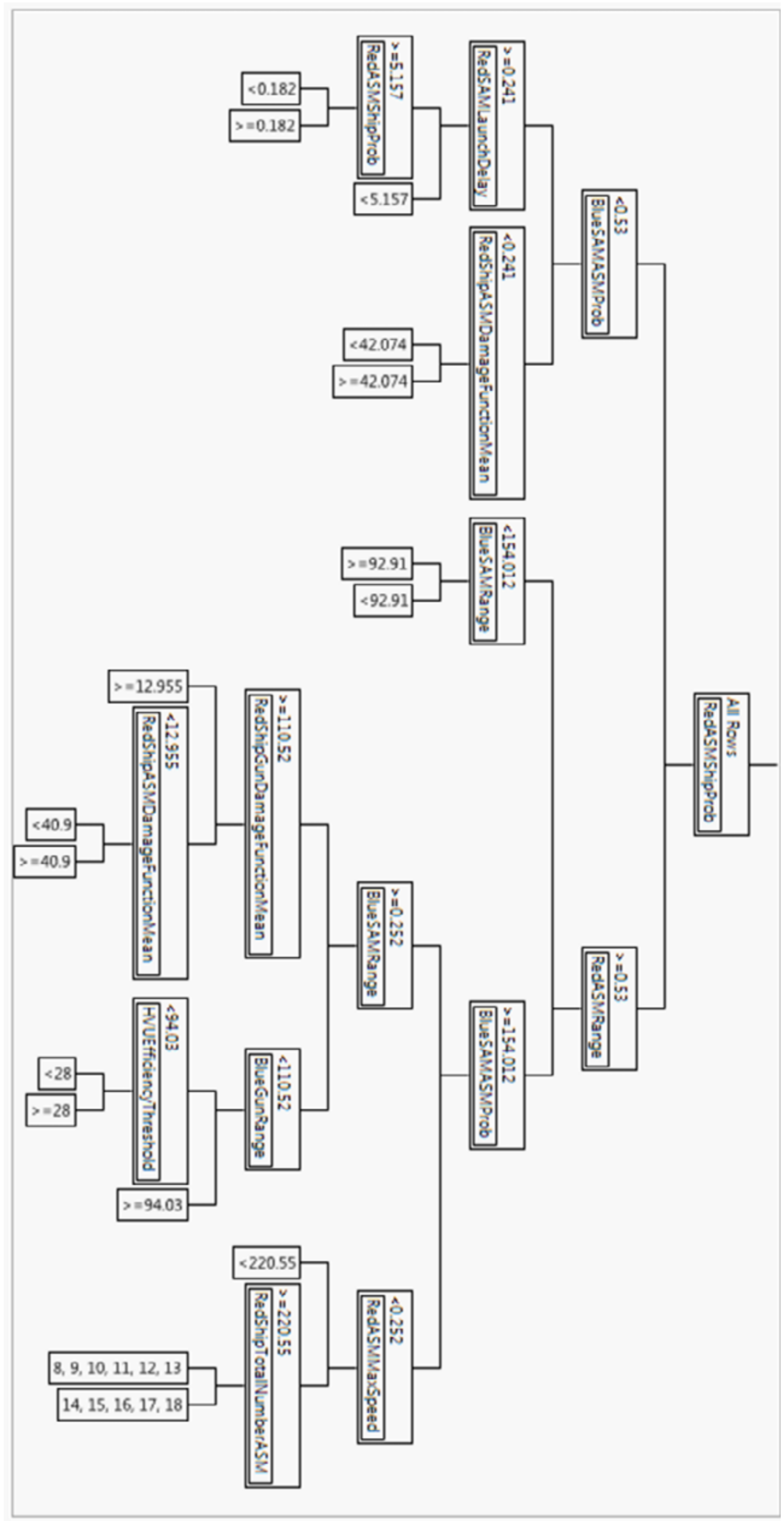


Figure 46. Small View of Partition Tree For Analysis of All Factors

The tree is first partitioned on Red ASM Blue ship probability of kill at the threshold of 0.53. On the left branch, the Blue SAM Red ASM probability of kill caused tree to partition at the threshold of 0.241. Down the branch, Red ASM launch delay and Red ASM damage function mean is effective on the HVU efficiency. Below Red ASM launch delay, Red ASM Blue ship probability of kill is effective on the efficiency of HVU at the threshold of 0.182.

The ROC curve that is obtained from this partition is seen Figure 47. The closer the value is to one, the better our model predicts the probability of HVU's survival. In other words, the greater area under the ROC curve, the better a tree model classifies the HVU's survival based on the factors that appear on the tree branches.

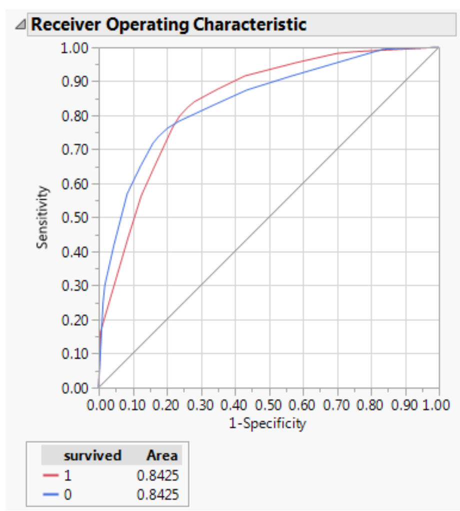


Figure 47. The ROC Curve from Partition Tree Analysis of All Factors.

The AUC for 15 splits is 0.8425. The column contributions, which shows the effect of each factor on each partitioning is seen in Figure 48. The factors that most affected the response are at the top.

Column Contributions

Term	Number of Splits	G^2	Portion
RedASMShipProb	2	103750.731	0.2989
BlueSAMASMProb	2	50785.601	0.1463
RedASMRange	1	38626.6401	0.1113
BlueSAMRange	2	38195.1639	0.1100
RedShipASMDamageFunctionMean	2	26218.5111	0.0755
BlueGunRange	1	17890.603	0.0515
RedShipGunDamageFunctionMean	1	15025.3648	0.0433
RedASMMaxSpeed	1	14857.8014	0.0428
HVUEfficiencyThreshold	1	14830.2105	0.0427
RedSAMLaunchDelay	1	14421.9116	0.0415
RedShipTotalNumberASM	1	12549.9699	0.0362

Figure 48. Column Contributions for Partition Tree Analysis of All Factors.

The number of splits is two on some factors, as seen in Figure 47. This shows that some of the variables may have a continuous effect and partition tree may not be the best analysis method for this data. So, our main analysis method is nominal logistic regression for this set of data. However, we can still have some insights about the model using the information in Figure 48.

The column contributions show the order of importance of the factors. As seen in Figure 48, the enemy ship ASMs blue ship probability of kill had the largest effect on the likelihood of the HVU's survival. The most important enemy ship properties other than that are the ASM range, the ASM damage mean after a successful hit, the gun damage mean after a successful hit, the ASM's maximum speed, the SAM's launch delay time, and the total number of ASMs. These results provide insight about effect of the enemy ships' ASM properties on the likelihood of HVU survival.

b. Nominal Logistic Regression

A nominal logistic regression is a regression model in which we have a binomial or binary response variable and any type of predictor variables. Logistic regression is the methodology that determines the relationship between the categorical dependent variables and one or more independent variables. We fit the model on a link, which is the logit function. For logistic regression, a link function is a placeholder for the response variable in least squares regression. In logistic regression, the model is fit over that link function instead of directly fitting on the response variable in a least squares regression. After the model is fit, the link function also provides insight into the likelihood of occurrence of the binary response variable. In our case, it is the HVU's survival.

In this section, a nominal regression is fit on the HVU's survival as a binary outcome. 101 factors are explanatory variables. 99 of them are the factors of the NOB design. Two of them are the number of Red ships and the number of Blue screen ships in each scenario.

After making a stepwise regression using minimum Bayesian Information Criterion (BIC) (Faraway, 2015), the nominal regression fit includes the variables presented in Figures 49 and 50.

Effect Summary

Source	LogWorth	PValue
Number of Red Ships	17559.68	0.00000
RedASMShipProb	16355.73	0.00000
BlueSAMASMProb	11232.12	0.00000
HVUEfficiencyThreshold	8341.225	0.00000
BlueSAMRange	8059.534	0.00000
RedASMRange	7525.562	0.00000
Number of Blue Ships	5926.434	0.00000
RedShipTotalNumberASM	4315.304	0.00000
RedShipASMDamageFunctionMean	3095.848	0.00000
BlueShipTotalNumberSAM	2889.394	0.00000
BlueclassificationTime	2795.507	0.00000
RedShipTotalNumberSAM	2661.897	0.00000
BlueShipTotalNumberASM	2297.567	0.00000
BlueCIWSASMProb	1610.869	0.00000
BlueShipCloseEnoughDistanceForImpactCIWS	1481.969	0.00000
BlueGunRange	1349.467	0.00000
BlueShipCloseEnoughDistanceForImpactGun	1293.600	0.00000
BlueShipMaxSpeed	869.202	0.00000
RedASMMaxSpeed	809.072	0.00000
BlueShipGunDamageFunctionSd	748.978	0.00000
RedShipEngagementSensorsDistortionSDOnXAxis	709.878	0.00000
BlueSAMLaunchDelay	687.170	0.00000
BlueNumberOfTimesToShootBeforeLook	681.220	0.00000
RedShipEngagementSensorsDistortionSDOnYAxis	678.428	0.00000
RedGunMaxDamage	649.289	0.00000
BlueASMMaxDamage	640.690	0.00000
HVUMaxSpeed	615.445	0.00000
BlueGunASMProb	611.270	0.00000
RedASMMaxDamage	552.311	0.00000
RedShipSurveillaceSensorDistortionSDOnXAxis	527.722	0.00000
RedGunRange	508.827	0.00000
RedShipCloseEnoughDistanceForImpactSAM	475.591	0.00000
BlueShipASMDamageFunctionMean	461.154	0.00000
RedShipCloseEnoughDistanceForImpactCIWS	407.363	0.00000
RedShipSurveillaceSensorDistortionMeanOnYAxis	358.966	0.00000
RedShipEfficiencyThreshold	354.974	0.00000
BlueASMRange	332.806	0.00000
BlueASMShipProb	319.705	0.00000
BlueShipCloseEnoughDistanceForImpactSAM	319.282	0.00000
BlueShipASMDamageFunctionSd	309.443	0.00000
BlueCIWSRange	307.543	0.00000
BlueGunShipProb	305.801	0.00000
RedDetectionRate	304.741	0.00000
RedNumberOfTimesToShootBeforeLook	304.679	0.00000
BlueShipCloseEnoughDistanceForImpactASM	283.492	0.00000
RedPolicy	273.630	0.00000
RedASMLaunchDelay	269.379	0.00000
BlueShipEfficiencyThreshold	224.648	0.00000
BlueGunMaxDamage	206.251	0.00000
BlueCIWSInterShootDelay	166.365	0.00000
BluegunInterShootDelay	163.058	0.00000
RedGunMaxSpeed	156.292	0.00000
BlueASMMaxSpeed	154.707	0.00000
BlueASMMinDamage	151.412	0.00000
RedShipCloseEnoughDistanceForImpactGun	118.718	0.00000
RedShipCloseEnoughDistanceForImpactASM	114.576	0.00000
BlueShipSurveillaceSensorDistortionSDOnYAxis	103.522	0.00000

Figure 49. Effect Summary of First Nominal Logistic Regression for All Factors.

Effect Summary

Source	LogWorth	PValue
RedGunShipProb	100.857	0.00000
RedASMMinDamage	93.751	0.00000
RedShipTargetCheckInterval	92.074	0.00000
RedCIWSInterShootDelay	83.647	0.00000
BlueDetectionRate	80.557	0.00000
BlueShipEngagementSensorsDistortionMeanOnYAxis	76.080	0.00000
RedSAMLaunchDelay	58.916	0.00000
RedSAMASMProb	58.669	0.00000
RedShipGunDamageFunctionMean	55.085	0.00000
BlueShipSurveillaceSensorDistortionMeanOnXAxis	53.593	0.00000
BlueShipTargetCheckInterval	52.211	0.00000
RedShipMaxSpeed	50.181	0.00000
RedShipASMDamageFunctionSd	44.724	0.00000
BlueGunMinDamage	43.943	0.00000
BlueShipGunDamageFunctionMean	40.531	0.00000
RedSAMMaxSpeed	40.452	0.00000
RedclassificationTime	37.121	0.00000
BlueShipSurveillaceSensorDistortionSDOnXAxis	35.042	0.00000
BlueGunMaxSpeed	32.719	0.00000
RedCIWSASMProb	30.535	0.00000
BlueShipTimeToDetect	28.944	0.00000
BlueShipSurveillaceSensorDistortionMeanOnYAxis	27.768	0.00000
RedGunASMProb	27.742	0.00000
RedGunMinDamage	27.257	0.00000
BlueCIWSMaxSpeed	27.149	0.00000
BlueSAMMaxSpeed	24.555	0.00000
HVUCIWSInterShootDelay	22.042	0.00000
HVUCloseEnoughDistanceImpactForCIWS	20.167	0.00000
BlueShipEngagementSensorsDistortionMeanOnXAxis	18.530	0.00000
RedgunInterShootDelay	15.489	0.00000
RedShipSurveillaceSensorDistortionSDOnYAxis	12.191	0.00000
RedShipEngagementSensorsDistortionMeanOnYAxis	7.552	0.00000
RedShipSurveillaceSensorDistortionMeanOnXAxis	5.769	0.00000
RedShipGunDamageFunctionSd	4.282	0.00005
BluePolicy	3.877	0.00013
RedCIWSRange	3.425	0.00038
BlueShipEngagementSensorsDistortionSDOnXAxis	3.184	0.00065
RedSAMRange	2.892	0.00128
HVUCIWSRange	2.167	0.00680
RedShipTimeToDetect	1.491	0.03227

Figure 50. Effect Summary of First Nominal Regression for All Factors (continued).

The model includes most of the factors that are statistically significant. However, most of these factors do not have practical significance inside the model. The ROC curve obtained from that fit and the AUC is seen in Figure 51.

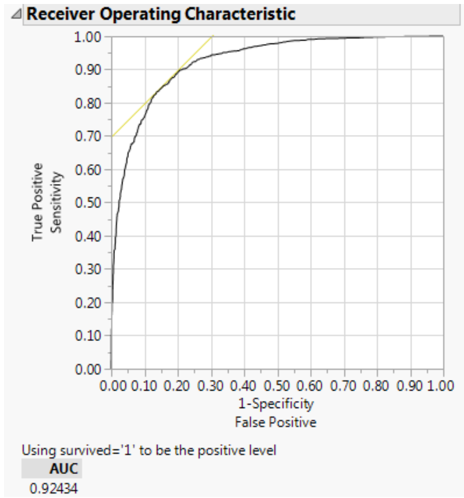


Figure 51. The ROC Curve for the First Nominal Logistic Regression for All Factors.

As seen in Figure 51, we have 0.92434 AUC if we include all of the factors that are seen in Figures 49 and 50. After removing practically insignificant factors, the effect summary of the model is seen in Figure 52.

Effect Summary

Source	LogWorth	PValue
RedASMShipProb	20546.94	0.00000
Number of Red Ships	16250.54	0.00000
BlueSAMASMProb	12102.83	0.00000
BlueSAMRange	8153.615	0.00000
HVUEfficiencyThreshold	8147.009	0.00000
RedASMRange	7416.666	0.00000
Number of Blue Ships	5178.975	0.00000
RedShipTotalNumberASM	5171.578	0.00000
BlueShipTotalNumberASM	3973.749	0.00000
RedShipASMDamageFunctionMean	3586.080	0.00000
RedShipTotalNumberSAM	2948.875	0.00000
BlueShipTotalNumberSAM	2653.663	0.00000
BlueShipCloseEnoughDistanceForImpactCIWS	2434.898	0.00000

Figure 52. Effect Summary of Last Nominal Regression for All Factors.

Statistically significant factors for the AAW Analysis Model are shown in Figures 49 and 50. Most of the factors in the AAW Analysis Model are statistically significant. With so many significant factors, a simulation is necessary to capture complexity. The factors that have practical significance are displayed in Figure

52. According to that reduced model, the Red ASM Blue ship probability of kill, number of red ships, Blue SAM Red ASM probability of kill, Blue ship SAM range and HVU efficiency threshold are among the most significant factors. The ROC curve obtained from the reduced model is seen in Figure 53.

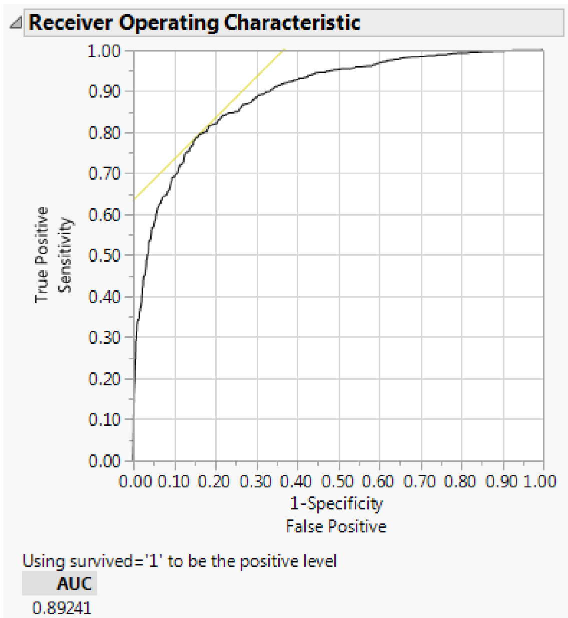


Figure 53. The ROC Curve for Last Nominal Logistic Regression for All Factors.

As seen in the figure, we only lose 3% from the AUC value by removing most of the factors that are practically insignificant, even if they are statistically significant. The prediction profiler shows the effect of the important factors on probability of HVU survival. The prediction profiler for the nominal regression model is seen in Figures 54, 55 and 56. The y-axis for the prediction profiler is only displayed on Figure 54.

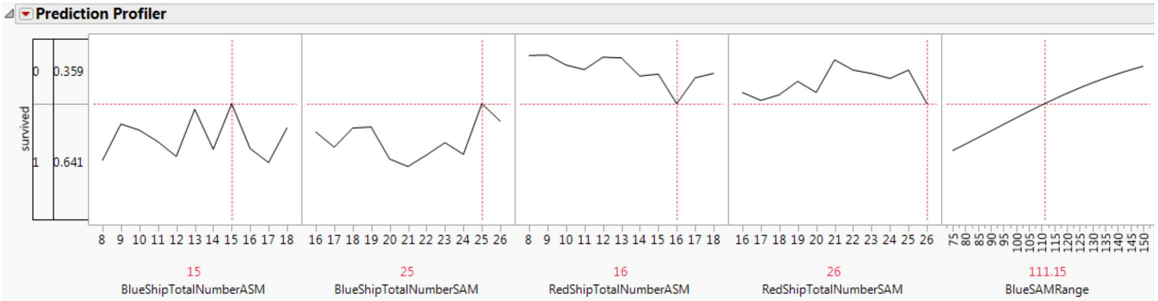


Figure 54. Prediction Profiler of Last Nominal Logistic Regression for All Factors.

Blue ship total number of ASMs profiler does not demonstrate a predictable pattern. However, as seen in the profiler, the likelihood of HVU survival reaches its maximum value when Blue ships have 15 ASMs. Also, the Blue ship total number of SAMs has a zigzagged effect on the likelihood of survival; whereas, the likelihood of survival is the maximum value when Blue ships had 25 SAMs. On the other hand, the probability of HVU survival decreases as the number of SAMs and ASMs of enemy ships increases. The likelihood of HVU survival is at the minimum level when Red ships have 16 ASMs and 26 SAMs. An increase in Blue SAM range increases the probability of survival, as it also increases the reaction time against Red ASMs. The zigzagged pattern at Blue ship total number of ASMs, SAMs, and Red ship total number of ASMs and SAMs stems from the fact that they are discrete variables and modeled as ordinal in statistical analysis using JMP.

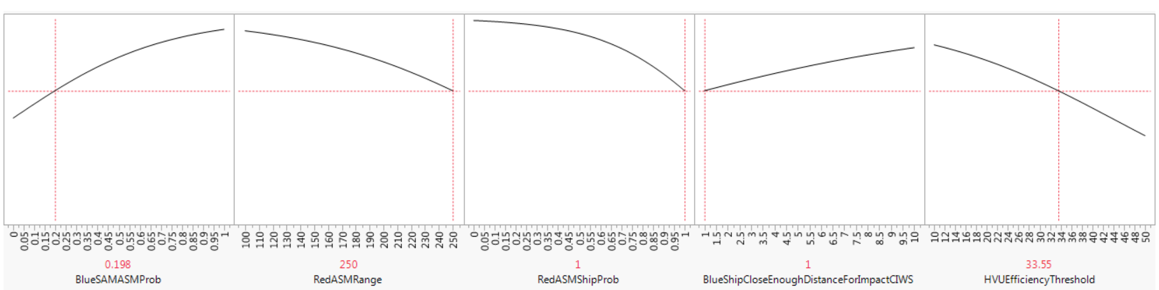


Figure 55. Prediction Profiler of Last Nominal Logistic Regression for All Factors (continued).

The increase in Blue SAM's Red ASM probability of kill also increases the HVU's probability of survival. On the contrary, as the Red ship's ASM range and Red ASM's Blue ship probability of kill increases, the likelihood of the HVU's survival decreases. The Close Enough Distance (CED) to impact for Close-In Weapon System (CIWS) rounds and gun rounds are the minimum distances required to trigger an impact event between the target and the particular gun round. As these distances increase, even if CIWS rounds or gun rounds fall far from their target, they impact. As the CED for impact of CIWS rounds increases, the likelihood of the HVU's survival increases. So, having more lethal rounds increases the probability of survival of the HVU. The HVU gets out of the battle if its efficiency gets below a user defined threshold. So, the higher this threshold, the less time the HVU stayed in the battle. The higher threshold may be interpreted as less armor or risk tolerance for the HVU. The lower the efficiency threshold, the more staying power that HVU has. So, the probability of survival increases as the efficiency threshold decreases.

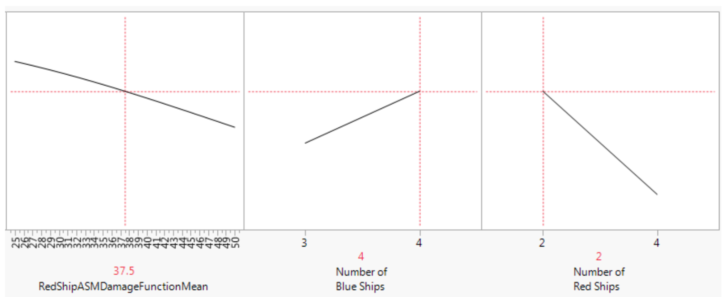


Figure 56. Prediction Profiler of Last Nominal Logistic Regression for All Factors (continued).

The Red ship ASM damage mean decreases the survival of the HVU, as expected. The more ASM damage a Red missile delivers, the less the likelihood of survival. Lastly, as analyzed in the overview section, increasing the number of Blue ships positively affects the likelihood of survival, whereas increasing the number of red ships has a negative effect.

c. *Results Summary*

We have determined the most important factors with both methodologies. Most of them were consistent with each other, even if not 100% in agreement. Depending on the column contributions in the partition tree and the effect summary in the nominal logistic regression, the most important factors are seen in Table 4.

Table 4. Results Summary for Most Important Factors Among All Factors.

Methodology	Most Important Factors
Partition Tree	***Red ASM Blue Ship Probability of Kill*** ***Blue Ship SAM Red ASM Probability of Kill*** ***Red Ship ASM Range*** ***Blue Ship SAM Range*** ***Red Ship ASM Damage Mean*** Blue Ship Gun Range Red Ship Gun Damage Mean Red Ship ASM Maximum Speed ***HVU Efficiency Threshold*** Red Ship SAM Launch Delay Time ***Red Ship Total Number of ASM***
Logistic Regression	***Red ASM Blue Ship Probability of Kill*** Number of Red Ships ***Blue Ship SAM Red ASM Probability of Kill*** ***Blue Ship SAM Range*** ***HVU Efficiency Threshold*** ***Red Ship ASM Range*** Number of Blue Ships ***Red Ship Total Number of ASM*** Blue Ship Total Number of ASM ***Red Ship ASM Damage Mean*** Red Ship Total Number of SAM Blue Ship Total Number of SAM Blue Ship CIWS CED for Impact of Rounds

As seen in Table 4, results of both models agree on seven of the factors. Among those seven factors, three of them are related to Blue ships and the HVU, four of them related to enemy ships. The most important factors of Blue ships on the HVU survival are ***Blue Ship SAM Red ASM Probability of Kill, Blue Ship***

SAM Range, and HVU Efficiency Threshold. The most effective factors of enemy ships are ***Red ASM Blue Ship Probability of Kill, Red Ship ASM Range, Red Ship Total Number of ASM, and Red Ship ASM Damage Mean.***

On the other hand, both models' results on analysis of all factors agrees that the enemy ships' ASM properties have the greatest effect on likelihood of the HVU's survival.

4. Analysis on Blue Ship Factors

a. Partition Tree

In this section, using the same partition tree methodology, the HVU's survival on Blue ship factors and aggregate output data from all three scenarios are analyzed. In other words, in this section, only controllable factors are analyzed.

While splitting the data, the ROC curve was continuously checked. The tree split below shows the split points depending on the response variables of Blue ship parameters. We have total of 52 factors of Blue ships, including HVU ship parameters—which is also a Blue ship, and one factor for the number of Blue screen ships in each scenario. To achieve, 0.80 AUC on the classification of data, 15 splits are required.

Small tree view of partition tree analysis is shown in Figure 57. Blue SAM Red ASM probability of kill is the most decisive factor for building partition tree. If we examine the left branch, It is seen that HVU efficiency threshold, Blue ship SAM launch delay, Blue SAM Red ASM probability, Blue ship SAM range, Blue ship ASM maximum damage, Blue ship surveillance sensor quality and Blue ship ASM launch delay of kill are the most significant factors. When we examine the left branch, it is seen that the tree is split on similar factors.

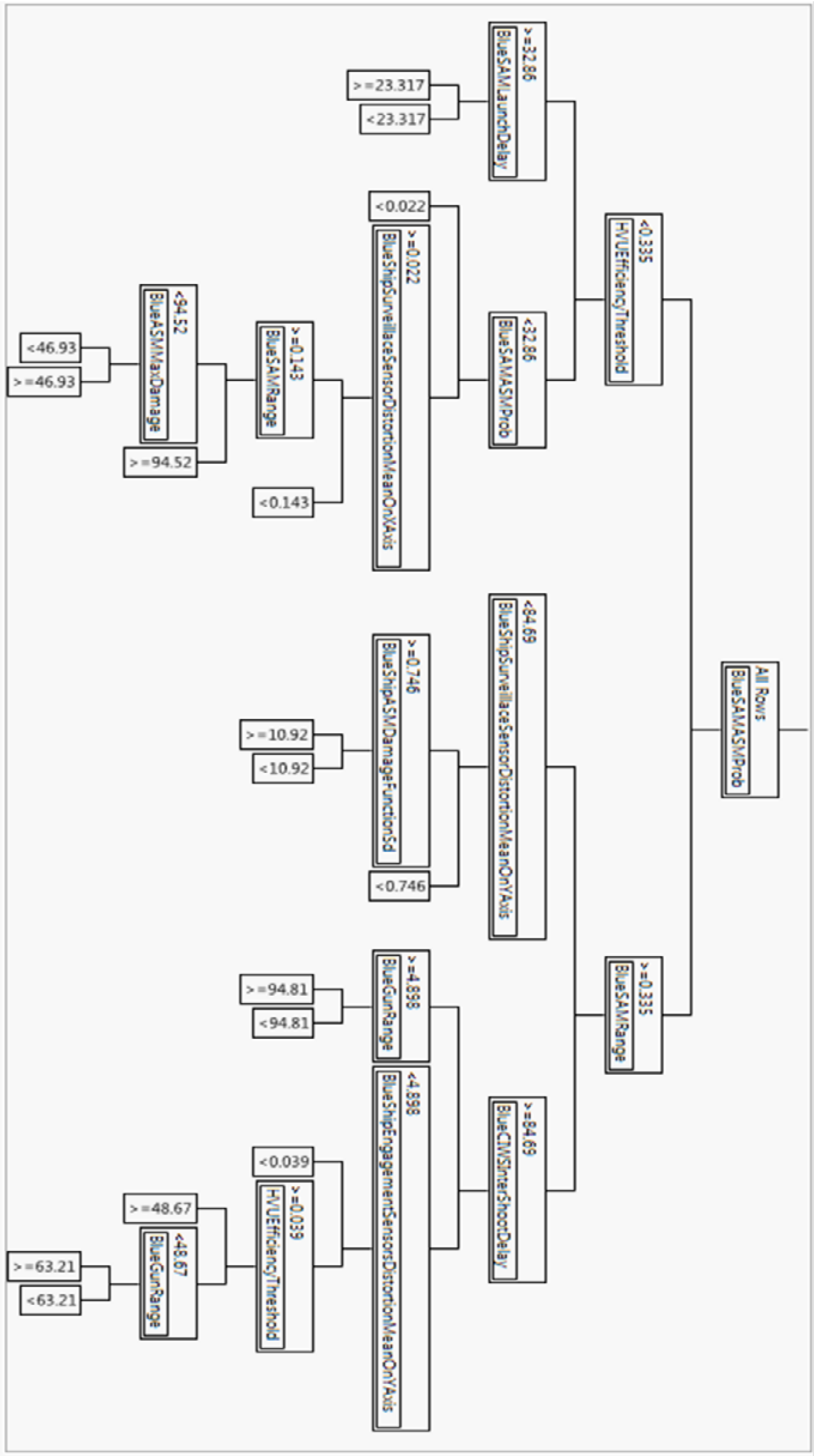


Figure 57. Small View of Partition Tree for Analysis of Blue Ship Factors.

The ROC curve obtained from the partition tree is shown in the Figure 58. As seen in the plot, the AUC is 0.8029.

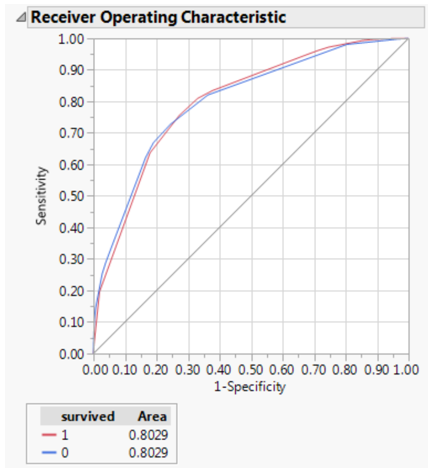


Figure 58. The ROC Curve for Partition Tree of Blue Ship Factors.

According to the partition tree of Blue ship factors data, the most important factors are seen in Figure 59.

Column Contributions

Term	Number of Splits	G^2	Portion
BlueSAMASMProb	2	60827.6419	0.2328
BlueSAMRange	2	50378.3673	0.1928
HVUEfficiencyThreshold	2	39369.2561	0.1506
BlueCIWSInterShootDelay	1	25509.9198	0.0976
BlueGunRange	2	20765.3273	0.0795
BlueShipEngagementSensorsDistortionMeanOnYAxis	1	15241.6272	0.0583
BlueShipSurveillaceSensorDistortionMeanOnYAxis	1	13514.277	0.0517
BlueASMMaxDamage	1	12635.1225	0.0483
BlueSAMLaunchDelay	1	11550.8209	0.0442
BlueShipSurveillaceSensorDistortionMeanOnXAxis	1	11547.3912	0.0442

Figure 59. Column Contributions of Partition Tree for Blue Ship Factors.

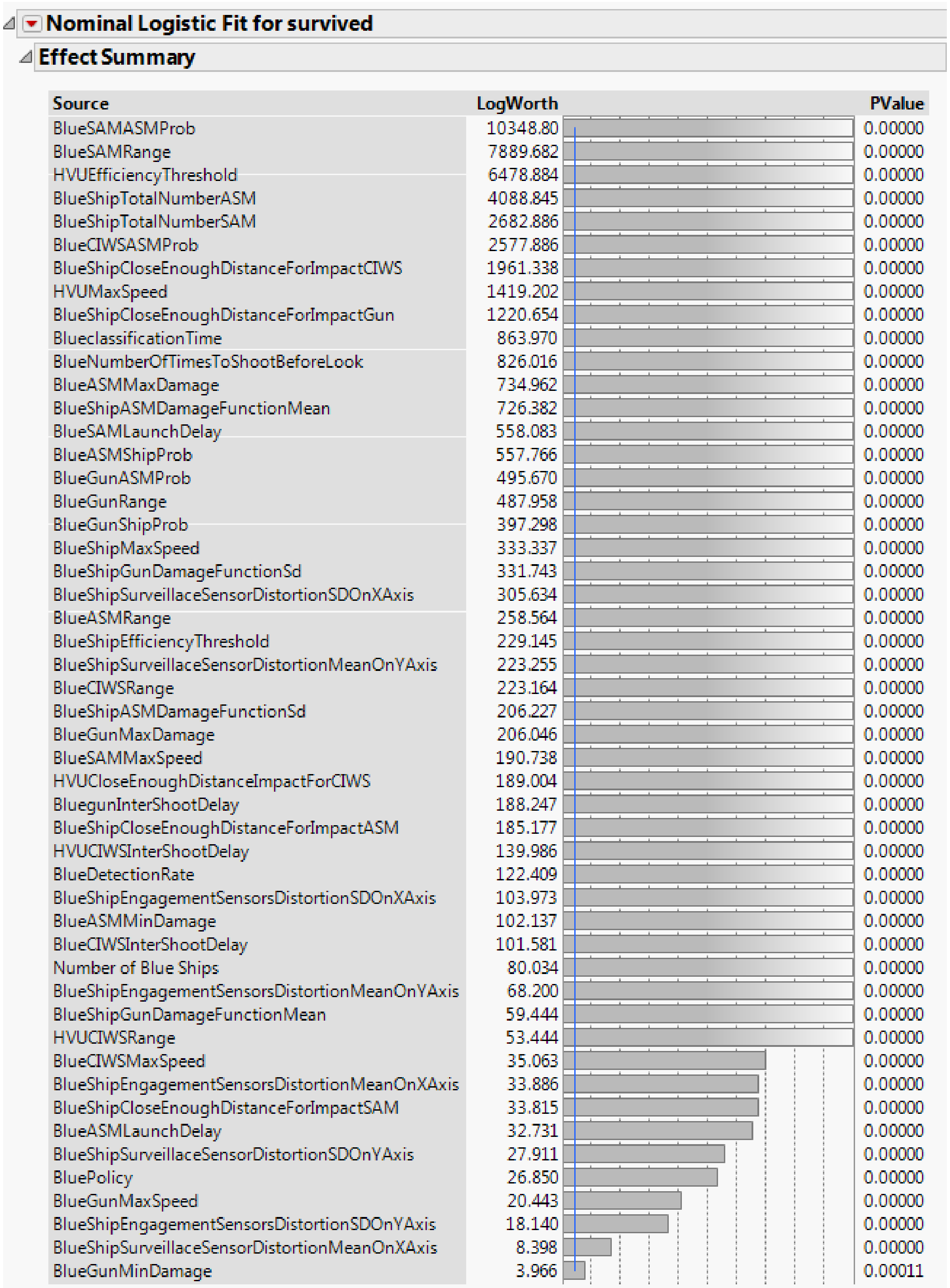
The number of splits is two on some factors shown in Figure 59. This also shows that the response variable may have a continuous effect, and the partition tree may not be the best analysis method for that data—as we encountered in the previous partition tree model. So, our main analysis method is again nominal logistic regression for this set of data. However, we can still have some insights about the model using the information in Figure 59.

According to the partition tree model, Blue ship's SAM Red ship ASM probability of kill, Blue ship SAM range, the HVU's efficiency threshold, which can be interpreted as the staying power of the HVU, Blue ships' CIWS rounds inter-shot delay time, Blue ship gun range, Blue ship engagement sensor quality, Blue ship ASM maximum range, Blue ships' SAM launch delay time, and Blue ship surveillance sensor quality are the most effective factors on the survival of HVU. The order of the importance is seen in Figure 59.

b. Nominal Logistic Regression

In this section, the nominal regression fit on the HVU's survival as a binary outcome is made again. 49 factors are explanatory variables. 47 of them are the factors of NOB design. Two of them are the number of Red ships and the number of Blue screen ships in each scenario.

After running a stepwise regression using minimum Bayesian Information Criterion (BIC) and fitting the model on the binary response variable HVU survival, statistically significant factors are as seen in Figure 60.



Nominal Logistic Fit for survived

Effect Summary

Source	LogWorth	PValue
BlueSAMASMProb	10348.80	0.00000
BlueSAMRange	7889.682	0.00000
HVUEfficiencyThreshold	6478.884	0.00000
BlueShipTotalNumberASM	4088.845	0.00000
BlueShipTotalNumberSAM	2682.886	0.00000
BlueCIWSASMProb	2577.886	0.00000
BlueShipCloseEnoughDistanceForImpactCIWS	1961.338	0.00000
HVUMaxSpeed	1419.202	0.00000
BlueShipCloseEnoughDistanceForImpactGun	1220.654	0.00000
BlueclassificationTime	863.970	0.00000
BlueNumberOfTimesToShootBeforeLook	826.016	0.00000
BlueASMMaxDamage	734.962	0.00000
BlueShipASMDamageFunctionMean	726.382	0.00000
BlueSAMLaunchDelay	558.083	0.00000
BlueASMShipProb	557.766	0.00000
BlueGunASMProb	495.670	0.00000
BlueGunRange	487.958	0.00000
BlueGunShipProb	397.298	0.00000
BlueShipMaxSpeed	333.337	0.00000
BlueShipGunDamageFunctionSd	331.743	0.00000
BlueShipSurveillaceSensorDistortionSDOnXAxis	305.634	0.00000
BlueASMRange	258.564	0.00000
BlueShipEfficiencyThreshold	229.145	0.00000
BlueShipSurveillaceSensorDistortionMeanOnYAxis	223.255	0.00000
BlueCIWSRange	223.164	0.00000
BlueShipASMDamageFunctionSd	206.227	0.00000
BlueGunMaxDamage	206.046	0.00000
BlueSAMMaxSpeed	190.738	0.00000
HVUCloseEnoughDistanceImpactForCIWS	189.004	0.00000
BluegunInterShootDelay	188.247	0.00000
BlueShipCloseEnoughDistanceForImpactASM	185.177	0.00000
HVUCIWSInterShootDelay	139.986	0.00000
BlueDetectionRate	122.409	0.00000
BlueShipEngagementSensorsDistortionSDOnXAxis	103.973	0.00000
BlueASMMinDamage	102.137	0.00000
BlueCIWSInterShootDelay	101.581	0.00000
Number of Blue Ships	80.034	0.00000
BlueShipEngagementSensorsDistortionMeanOnYAxis	68.200	0.00000
BlueShipGunDamageFunctionMean	59.444	0.00000
HVUCIWSRange	53.444	0.00000
BlueCIWSMaxSpeed	35.063	0.00000
BlueShipEngagementSensorsDistortionMeanOnXAxis	33.886	0.00000
BlueShipCloseEnoughDistanceForImpactSAM	33.815	0.00000
BlueASMLaunchDelay	32.731	0.00000
BlueShipSurveillaceSensorDistortionSDOnYAxis	27.911	0.00000
BluePolicy	26.850	0.00000
BlueGunMaxSpeed	20.443	0.00000
BlueShipEngagementSensorsDistortionSDOnYAxis	18.140	0.00000
BlueShipSurveillaceSensorDistortionMeanOnXAxis	8.398	0.00000
BlueGunMinDamage	3.966	0.00011

Figure 60. Effect Summary of First Nominal Logistic Regression of Blue Ship Factors.

However, most of the variables' log worth values are smaller by an order of magnitude compared to the most important factor, which is the Blue ship's

SAM Red ASM probability of kill. So, they are statistically significant, but they may not have much practical significance. The ROC curve, which also shows the quality of the model for the nominal logistic regression is seen in Figure 61, here we include all of the statistically significant factors.

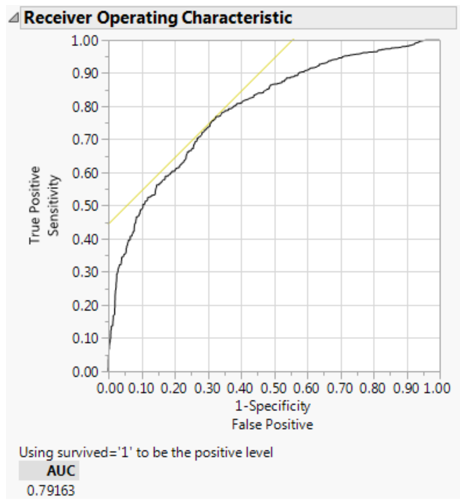


Figure 61. The ROC Curve for the First Logistic Regression for Blue Ship Factors.

The AUC is 0.79163, as seen in the figure. However, the model is not useful if it includes most of the factors. So, the factors that are not practically significant have been removed from the model, giving us the more parsimonious model factors seen in Figure 62.

Effect Summary

Source	LogWorth	PValue
BlueSAMASMProb	9346.965	0.00000
BlueSAMRange	8008.619	0.00000
HVUEfficiencyThreshold	6307.711	0.00000
BlueShipTotalNumberASM	4590.176	0.00000
BlueCIWSASMProb	2622.231	0.00000
BlueShipTotalNumberSAM	2187.411	0.00000
BlueShipCloseEnoughDistanceForImpactCIWS	2098.709	0.00000
BlueShipCloseEnoughDistanceForImpactGun	1595.668	0.00000
HVUMaxSpeed	1236.058	0.00000

Figure 62. Effect Summary of the Last Nominal Logistic Regression of Blue Ship Factors.

Most of the factors in Figure 62 are consistent with the ones which we obtained in the partition tree methodology. The ROC curve for the model is displayed in Figure 63.

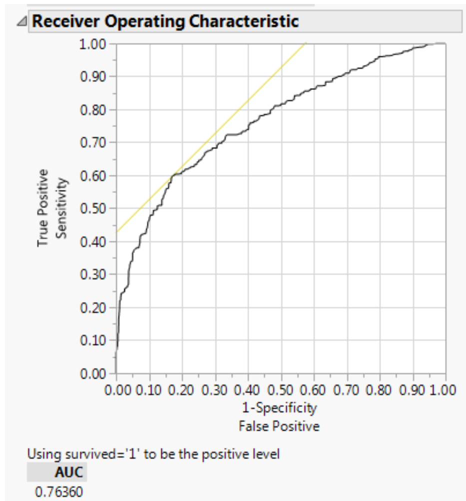


Figure 63. The ROC Curve of the Last Logistic Regression Blue Ship Factors.

As seen in Figure 63, we only lose 3% of our AUC value, after taking out all the practically insignificant predictors. Figures 64 and 65 demonstrate the prediction profiler plots of the logistic fit, which show the effect of predictors on the response variable. The y-axis for the prediction profiler is only displayed on Figure 64. In the nominal logistic regression, our model predicts the probability of survival.

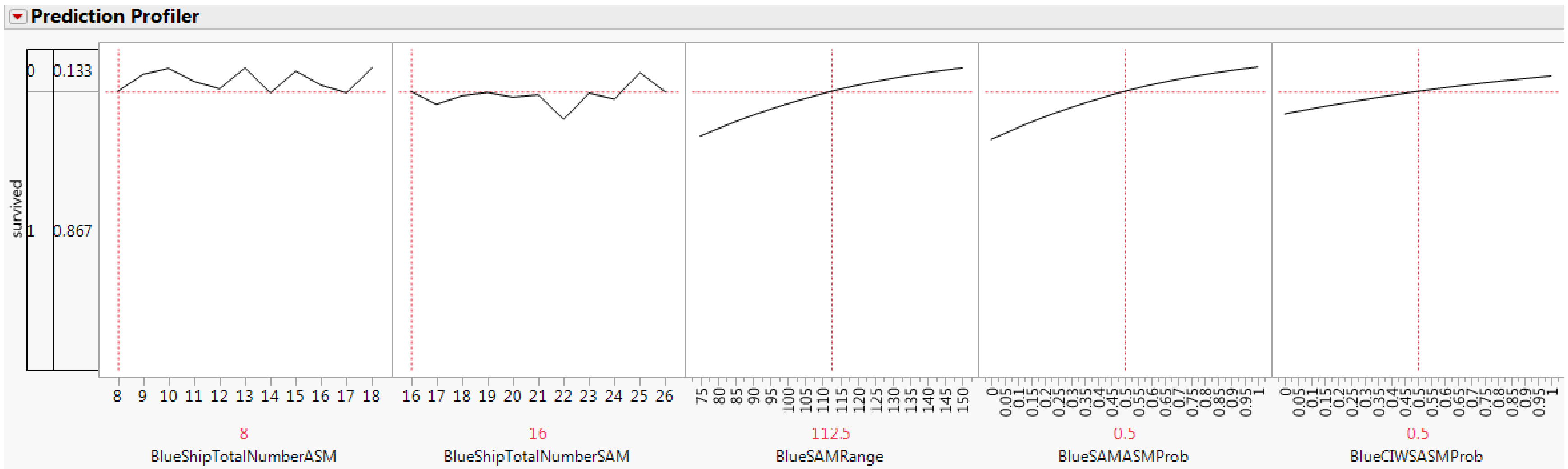


Figure 64. Prediction Profiler for Logistic Regression of Blue Ship Factors.

The total number of ASMs at each Blue ship is an effective factor. Even if there is not a clear pattern how it affects the probability of the HVU's survival, the HVU's survival probability is highest if Blue ships have the maximum number of ASMs. The total number of SAMs at each Blue ship has the same effect. The likelihood of survival is maximized at 25 SAMs for this scenario. The zigzagged pattern at Blue ship total number of ASMs and SAMs stems from the fact that they are discrete variables and modeled as ordinal in statistical analysis using JMP. The Blue ship SAM range, the Blue ship's SAM Red ASM probability of kill and the Blue ship's CIWS Red ASM probability of kill also have a very significant effect on the likelihood of the HVU's survival. As they increase, the probability of survival also increases.

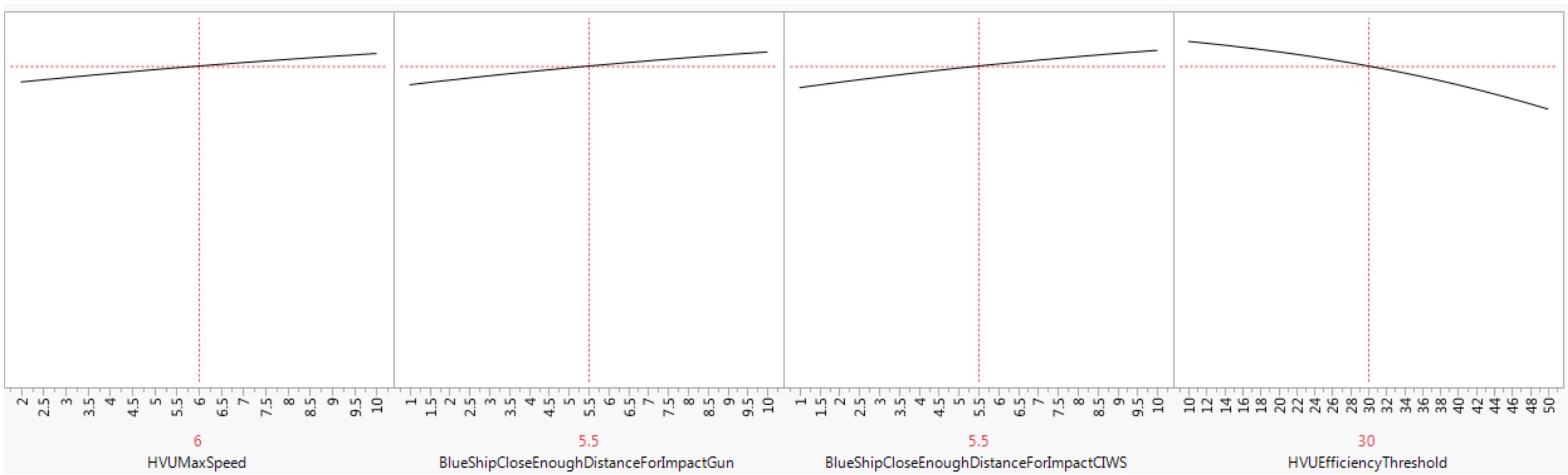


Figure 65. Prediction Profiler for Logistic Regression of Blue Ship Factors (continued)

The HVU's maximum speed increased the likelihood of survival. This can be interpreted as the effect of a decrease in exposure time to threats. An increase in the CED to impact for CIWS rounds and gun rounds also increases the probability of survival of the HVU. An increase in the HVU efficiency threshold decreases the probability of survival as aforementioned in the previous analysis.

c. Results Summary

We have determined the most effective factors among Blue ship properties with both methodologies. Most of them are consistent with each other, even if not 100% in agreement. The most important factors are shown in Table 5 depending on the analysis methodology.

Table 5. Results Summary for Most Important Factors Among Blue Ship Factors.

Methodology	Most Important Factors
Partition Tree	***Blue Ship SAM Red ASM Probability of Kill*** ***Blue Ship SAM Range*** ***HVU Efficiency Threshold*** Blue CIWS Inter Shoot Delay Time Blue Gun Range Blue Ships Engagement Sensor Quality Blue Ships Surveillance Sensor Quality Blue Ships ASM Maximum Damage Blue Ships SAM Launch Delay Time
Logistic Regression	***Blue Ship SAM Red ASM Probability of Kill*** ***Blue Ship SAM Range*** ***HVU Efficiency Threshold*** Blue Ship Total Number of ASM Blue Ship CIWS Red ASM Probability of Kill Blue Ship Total Number of SAM Blue Ship COD for Impact of CIWS Rounds Blue Ship COD for Impact of Gun Rounds HVU Maximum Speed

As seen in the table, results of both models agreed on the SAM properties and the HVU's staying power, so among all these factors that we can control, the most important three are assessed as ***Blue Ship's SAM Red ASM Probability of Kill, Blue Ship's SAM Range*** and ***HVU Efficiency Threshold*** according to the analysis of Blue ship factors.

The model results on Blue ship factors (only controllable) and the models on all factors (both controllable and uncontrollable) including Red ship factors agree that SAM properties and HVU staying power are the most effective factors on likelihood of HVU survival.

V. CONCLUSIONS AND FUTURE WORK

A. CONCLUSIONS

Convoy operations in naval combat missions are significant and may have crucial effects on an overall campaign's success. The main purpose of convoy operations in naval tactics is the protection of an HVU using surface, air, and underwater assets. Threats in a convoy operation may also stem from underwater, surface, and air assets. The AAW threat is one of the primary concerns in convoy operations.

In this study, the scope of convoy operations was narrowed down. Protection of HVU was modeled with screen ships, along with their sensors and weapons. The threat environment was modeled with enemy surface assets, and the weapons and sensors they have. The model was built in a generic manner such that screen ships and enemy ships, sensors, and weapons can be instantiated in any configuration. The Microsoft Access database was used to specify characteristics of each ship, and also the weapons and sensors that each ship has. Microsoft Access database serves as a GUI for instantiation of each scenario.

The concept of operations, such as positioning of screen ships, the layered defense policy, and SAM shooting policies were also modeled inside the AAW Analysis Model. The modeling methodology was DES, and the implementing tool was the Simkit library of the Java programming language. Each component and structure of the AAW Analysis Model was summarized in Chapter III.

Air threats may also stem from the ASMs that are launched from land based sites and airplanes. However, neither of them was modeled inside this study. Also, eliminating air threats can be done by soft kill methods, such as decoys and electronic warfare. In the AAW Analysis Model, only hard kill methods were modeled.

MOEs were determined to be the efficiency of the HVU at the end of the simulation and the binary outcome of whether the HVU survives or not. Before starting the simulation, a starting position, a goal, and a path were defined for the HVU. Screen ships protected the HVU from their relative positions according to the HVU's position. If the HVU survived until the predefined goal was reached, the operation was assumed to be successful. MOPs were defined as the characteristics of Blue ships, as they affected the outcome.

In Chapter IV, a detailed analysis on both controllable and uncontrollable factors was made. A total of 99 factors were analyzed in the AAW Analysis Model. 52 of factors were controllable and 47 of were uncontrollable. The controllable factors consisted of screen ship and HVU characteristics, whereas the uncontrollable factors were properties of enemy ships that were posing a threat in the operational area.

Among these 99 factors, two of them were nominal, six of them were ordinal categorical variables, and 91 of them were continuous variables. To have a favorable correlation among these factors, a NOB LH design (Vieira, 2012) was used. This design led to a 3.56% absolute maximum pairwise correlation among factors. This design had 512 design points. Each design point included all factors input combinations.

We set up three scenarios varying the number of screen ships and the number of Red ships. For each simulation run, the simulation terminated when the HVU either reached the goal using its predefined path or was destroyed by any enemy ship at any time during the simulation. The simulation was run 512,000 times for each of three scenarios. We made a total of 1,536,000 simulated AAW battles to explore the most effective factors on convoy operations under air threat using the AAW Analysis Model. Each run took three milliseconds on average using a personal computer, which has Intel (R) Core (TM) i7–4810MQ 2.8 Ghz CPU and 8 GB RAM. Run times varied depending on the number of ships instantiated.

After obtaining the outputs and recording them in a Comma Separated Values (CSV) file, we compared the mean efficiency and survival rate of the HVU for each scenario and an overall analysis. The comparison showed that increasing the number of screen ships increased the HVU's survival probability, whereas more enemy ships decreased it.

Nominal logistic regression and partition tree methodologies were used to analyze the outcome of three scenarios. The outcomes from each scenario were aggregated to make the overall analysis. Two different analyses were made to capture all the possible insights that can be gained from the model. The first analysis included only controllable factors. On the other hand, the second analysis included both controllable and uncontrollable factors.

Both analysis methods were made on the binary response variable of HVU survival. Analysis of the controllable factors included controllable factors as explanatory variables, whereas analysis of all factors included both uncontrollable and controllable factors as explanatory variables. Analysis of controllable factors led to a robust design as the analysis collapsed the uncontrollable factors. Robust designs help us gain the insights into the model when we also include the uncontrollable factors in a DOE.

The analysis of uncontrollable factors showed that SAM specifications of screen ships and the staying power of the HVU were the most effective factors among the controllable ones, whereas the analysis of all factors showed that ASM specifications of enemy ships were the most impactful on the outcome of the HVU's survival. The most effective factors and their effects on the survival probability of the HVU as they increase are shown in Table 6.

Table 6. Most Effective Factors of AAW Analysis Model.

Factor	Type	Effect
Blue Ship SAM Red ASM Probability of Kill	Controllable	Positive
Blue Ships SAM Range	Controllable	Positive
HVU Efficiency Threshold	Controllable	Positive
Red ASM Blue Ship Probability of Kill	Uncontrollable	Negative
Red Ship ASM Range	Uncontrollable	Negative
Red Ship ASM Damage Mean	Uncontrollable	Negative
Red Ship Total Number of ASM	Uncontrollable	Negative

Including the uncontrollable factors in the analysis led to insights about the most effective uncontrollable factors on the success of convoy operation, which are highlighted in Table 6. This information is crucial and having this before starting a convoy operation may yield a better understanding of the vulnerabilities to enemy assets.

While doing the analysis, partition trees were found not to be appropriate for the analysis of AAW Model because the response variable of HVU survival had a continuous effect on the metamodel. However, the insights gained while implementing the partition tree analysis were utilized to conclude about the most effective factors in the AAW Analysis Model.

The interactions between ASMs and SAMs were decisive on the success of a convoy operation under air threat. These results stressed the importance of soft kill methods, such as usage of decoys and electronic warfare, which may be effective in decreasing the probability of successful hit for an ASM.

We can conclude that for screen ship designs we first need to focus on the specifications of SAMs we have. In other words, if we have a limited budget for a research and development for screen ships whose primary purpose is AAW, we first need to develop SAM properties. These properties may include, but are not limited to, SAM speed, successful hit probability, and range. Additionally, before starting a convoy operation the most crucial intelligence about the enemy is ASM specifications of enemy ships. As aforementioned, another critical research and development area is the electronic warfare concepts which are not modeled in this study, because the results of the study demonstrated that decreasing the successful probability of ASMs had one of the most impactful effects on the probability of the HVU's survival.

B. FUTURE WORK

The AAW Analysis Model is a model of moderate complexity, resolution, and fidelity. Even if it serves as a basis for further high fidelity models, it has some limitations that are stated in the assumptions. To have a better model, the ideas that are stated below may be incorporated into the AAW Analysis Model.

- Movements are modeled in linear motion. In real life, the movement of ships, gun rounds, and missiles are non-linear. Introducing non-linearity may increase the fidelity.
- Misclassification is ignored. However, in real operations, it is possible and may have drastic results on the outcome of battles. So, it needs to be introduced for better analysis.
- Since the AAW Analysis Model includes 99 factors, and objects with many parameters, a graphical user interface (GUI) may be useful for making it more user friendly.
- Evasive maneuvers among opposing or allied units are ignored. However, these can be very significant in real operations. According to doctrinal needs, they may also be introduced.
- Soft kill methods are ignored in this study, but inclusion would increase the fidelity of future studies.
- Air threats may also stem from fighter airplanes and land based missile sites. So they may also be introduced.
- For exploring tactics in a particular geographic operational area, integration of a geographic information system may be useful. While integrating geographic information systems, inclusion of meteorological effects on operations needs to be considered.
- The AAW Analysis Model results are highly dependent on the input combinations, ranges, and number of ships instantiated from both sides. More insights can be captured with different inputs and initial scenario combinations.

- Sensors in the AAW Analysis Model are either modeled using a constant rate or a cookie cutter approach, introducing a distortion factor dependent on the relative distance and offset angle of contact. Better sensor modeling approaches can be introduced.

- Rather than actual data—which is classified—open source data was used for the AAW Analysis Model's input factors. The model can be run using real data in a classified environment to obtain results that are closely aligned with actual systems.

LIST OF REFERENCES

Al Rowaei, A. (2011). The effect of time advance mechanism (TAM) in modeling and simulation: DES & DTS comparison analysis, case study: combat simulation. Doctoral dissertation. Naval Postgraduate School, Monterey,CA.

Armstrong, M. J. (2005). Stochastic salvo model for naval surface combat model. Informs, 53, 830–841. Retrieved from http://pubsonline.informs.org/doi/abs/10.1287/opre.1040.0195?journalCode=opre

Aydin, E. (2000). Screen dispositions of naval task forces (Master's thesis). Retrieved from http://calhoun.nps.edu/bitstream/handle/10945/7785/screendispositio00aydi.pdf?sequence=3&isAllowed=y

Buss, A. H. (2002). Component based simulation modeling with Simkit. In: *Proceedings of 2002 Winter Simulation Conference*, E. Yucesan, C.Chen, L.Snowdon, and J. Charnes (eds.). Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc., pp. 243–249.

Buss, A. H. (2011). Lecture on discrete event simulation modeling. Personal collection of A. H. Buss. Naval Postgraduate School, Monterey CA.

Buss, A. H. (2016). Simkit (Version 1.4.4) [Computer software]. Retrieved from https://eos.nps.edu/simkit/

Buss, A. H., & Sanchez, P. J. (2002). Building complex models with LEGO (listener event graph objects). In: *Proceedings of the 2002 Winter Simulation Conference*, E. Yucesan, C.Chen, L.Snowdon, and J. Charnes (eds.). Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc., pp. 243–249.

Buss, A. H., & Sanchez, P. J. (2005). Simple movement and detection in discete event simulation. In: *Proceedings of 2005 Winter Simulation Conference,* M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines (eds.). Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc., pp. 992–1000.

Cioppa, T. M., & Lucas, T. W. (2007). Efficient nearly orthogonal and space-filling Latin hypercubes. *Technometrics, 49*(1), 45–55.

Faraway, J. J. (2015). *Linear models with R*. Boca Raton, FL: Chapman & Hall/CRC.

Hernandez, A., Lucas, T. W., & Carlyle, M. (2012). Constructing nearly orthogonal Latin hypercubes for any nonsaturated run-variable combination. *ACM Transactions on Modeling and Computer Simulation, 22*(4), Article 20.

Kulac, O. (1999). A comparative analysis of active and passive sensors in anti-air warfare area defense using discrete event simulation components (Master's thesis). Retrieved from https://calhoun.nps.edu/bitstream/handle/10945/13620/99MarKulac.pdf?sequence=3&isAllowed=y

Law, A. M., & Kelton, W. D. (1991). *Simulation modeling and analysis.* New York: McGraw-Hill.

Lucas, T. W. (2000). The stochastic versus deterministic argument for combat simulations: Tales of when the average won't do. *Military Operations Research, 5*(3), 9–28.

Nakayama, M. K. (2008). Statistical analysis of simulation output. In: *Proceedings of the 2008 Winter Simulation Conference*, Scoot Mason, Ray Hill, Lars Monsh, Oliver Rose (eds.). Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc., pp. 66–72.

O'Neil, W. D. (1981). Naval anti-air warfare. *Journal of American Defense Preparedness Association*, 385.

Sanchez, P. J. (2007). Fundamentals of simulation modeling. In: *Proceedings of the 2007 Winter Simulation Conference*, S. G. Henderson, B. Biller, M. H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton (eds.). Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc., pp. 54–62.

Sanchez, S. M., & Wan, H. (2012). Work smarter not harder: A tutorial on designing and conducting simulation experiments. *Proceedings of the 2012 Winter Simulation Conference.* C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher (eds.). Piscataway, NJ: Institute of Electrical and Electronics Engineers, Inc., pp. 1795–1809.

SAS Institute Inc. (1989). JMP (Version 12.0) [Computer software], Cary, NC.Retrieved from https://www.nps.edu/Technology/SoftwareLib/Auth/PackageInfo.asp?pkg=jmpwin12

Scruben, L. (1983). Simulation modeling with event graphs. *Communications of the ACM, 26*(11), 957–963. Ithaca, NY: Cornell University. Retrieved from https://dl.acm.org/citation.cfm?id=358460

Townsend, J. R. (1999). Defense of naval task forces from anti-ship missile attack (Master's thesis). Retrieved from https://calhoun.nps.edu/bitstream/handle/10945/13646/99Mar_Townsend.pdf?sequence=3&isAllowed=y

Turan, B. (2002). A comperative analysis of ship-self air defense (SSAD) systems using a modkit simulation (Master's thesis). Retrieved from https://calhoun.nps.edu/bitstream/handle/10945/13647/99Mar_Turan.pdf?sequence=3&isAllowed=y

U.S.Navy. (n.d.). Retrieved May 3, 2016, from U.S.Navy: http://www.public.navy.mil/

Vieira, H. J. (2012). Naval Postgraduate School simulation experiments and efficient designs center for data farming . Retrieved February 17, 2016, from NOB_Mixed_512DP_v1.xlsx design spreadsheet: https://harvest.nps.edu/

Vieira, J. H., Sanchez , S. M., Kienitz, K. H., & Beldarrain, M. C. (2013). Efficient nearly orthogonal and balanced mixed designs: an effective way to trade-off analysis via simulation. *Journal of Simulation, 7*, 264–275. Retrieved from http://calhoun.nps.edu/bitstream/handle/10945/44877/Sanchez_Efficient_2013.pdf?sequence=4&isAllowed=y

Wackerly, D. D., Mendenhall, W., & Scheaffer, R. L. (2008). *Mathematical statistics with applications.* Belmont, CA: Cengage Learning.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California