NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

PLAN VALIDATION USING DES AND AGENT-BASED SIMULATION

by

Teck Hwee Wong
Kim Soo Ong

December 2008

Thesis Advisor: Christian J. Darken
Co-Advisor: Arnold H. Buss

Approved for public release; distribution is unlimited
This thesis was done at the MOVES Institute

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** December 2008	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Plan Validation Using DES and Agent-based Simulation		**5. FUNDING NUMBERS**	
6. AUTHOR(S) Teck Hwee Wong & Kim Soo Ong			
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** N/A	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited		**12b. DISTRIBUTION CODE**	

13. ABSTRACT (maximum 200 words)

Military plan validation is typically a long-drawn process requiring planners to validate their plans using anticipated scenarios or through military exercises. While military exercises provide realistic simulation of the plan, it is often the most expensive way of validating a plan. On the other hand, although using anticipated scenarios is relatively cheaper, the robustness of the validated plans is dependent on the extensiveness of the scenarios that they are validated against.

This thesis explores the possibility of using a multi-agent system (MAS) to generate the aggressor's air strike plans, which could be coupled with a low resolution Discrete Event Simulation (DES) based air defense simulator to augment human planners in their plans validation. A MAS-based strike plan generator was built based on the tactics described in air strike doctrines. A DES-based air defense simulator was also built to provide an agent environment by modeling the behavior of air defense assets and their interactions with the aggressor's fighters.

The resulting system demonstrates the ability to validate air defense plans using MAS generated strike plans and a low resolution DES-based simulator. It also provides a platform to assist air defense planners in foreseeing the action, reaction and counteraction dynamics of their air defense plan against a range of possible strike plans.

14. SUBJECT TERMS Discrete-Event Simulation, Agent-based, Plan Validation, Air Defense			**15. NUMBER OF PAGES** 115
			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

PLAN VALIDATION USING DES AND AGENT-BASED SIMULATION

Teck Hwee Wong
Civilian, Defence Science and Technology Agency, Singapore
B.Eng. (Electrical and Electronic Engineering), University of Leicester, 1998

Kim Soo Ong
Civilian, Defence Science and Technology Agency, Singapore
M.Tech. (Knowledge Engineering), National University of Singapore, 2005

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MODELING, VIRTUAL ENVIRONMENTS, AND SIMULATION (MOVES)

from the

NAVAL POSTGRADUATE SCHOOL
December 2008

Authors: Teck Hwee Wong
Kim Soo Ong

Approved by: Christian J. Darken
Thesis Advisor

Arnold H. Buss
Co-Advisor

Mathias Kölsch
Chairman, MOVES Academic Committee

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Military plan validation is typically a long-drawn process requiring planners to validate their plans using anticipated scenarios or through military exercises. While military exercises provide realistic simulation of the plan, it is often the most expensive way of validating a plan. On the other hand, although using anticipated scenarios is relatively cheaper, the robustness of the validated plans is dependent on the extensiveness of the scenarios that they are validated against.

This thesis explores the possibility of using a multi-agent system (MAS) to generate the aggressor's air strike plans, which could be coupled with a low resolution Discrete Event Simulation (DES) based air defense simulator to augment human planners in their plan validation. A MAS-based strike plan generator was built based on the tactics described in air strike doctrines. A DES-based air defense simulator was also built to provide an agent environment by modeling the behavior of air defense assets and their interactions with the aggressor's fighters.

The resulting system demonstrates the ability to validate air defense plans using MAS generated strike plans and a low resolution DES-based simulator. It also provides a platform to assist air defense planners in foreseeing the action, reaction and counteraction dynamics of their air defense plan against a range of possible strike plans.

THIS PAGE INTENTIONALLY LEFT BLANK

DISCLAIMER

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

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ABBREVIATIONS, ACRONYMS, AND SYMBOLS

AA	Anti-Air
AAA	Anti-Air Artillery
AAG	Anti-Air Gun
ADA	Air Defense Artillery
AI	Artificial Intelligence
AOR	Area of Responsibility
BRL	Bomb Release Line
DAFS	Dynamic Allocation of Fires and Sensors
DES	Discrete Event Simulation
DMS	Decision Making System
DTED	Digital Terrain Elevation Data
LEGO	Listener Event Graph Objects
MAJ	Major
MAS	Multi-Agent System
MOE	Measure of Effectiveness
MT	Mersenne Twister
NOLH	Nearly Orthogonal Latin Hypercube
Pk	Probability of kill
POC	Proof-of-Concept
SAM	Surface to Air Missile
SEAD	Suppression of Enemy Air Defenses
SHORAD	Short Range Air Defense
SME	Subject Matter Expert
SSKP	Single Shot Kill Probability
UDP	User Datagram Protocol
XML	Extensible Markup Language

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

The authors would like to acknowledge Professor Christian Darken and Professor Arnold Buss for their expert guidance and unwavering patience. Their assistance on designing the agent's behavior and model structuring was invaluable. In addition, the authors would also like to thank MAJ Teo Hoon Hong (Crafty) and MAJ Lee Kok Kiang from the Singapore Armed Forces for sharing their experts' knowledge in air strike and air defense operations.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

Much of the military planning, whether it is offensive or defensive, is based on the expected adversary course of action, tactics and doctrine. If the adversary manages to produce a course of action which is not anticipated in the plan, this will lead to the adversary gaining a tactical advantage, and in the worst case, allow the adversary a strategic edge.

In today's military planning, generated military plans are often difficult to validate and verify, and unless they are actually put to use in a real scenario, it is hard to gauge the actual effectiveness of the plan.

This thesis will focus on incorporating an agent-based simulation using discrete events. It will explore the effect of dynamic decision making aspect of the adversary agents based on its current environment. The second objective of the thesis is to develop a test-bed simulation to evaluate the effectiveness of the adversary agents and to evaluate the robustness of the air defense plans.

B. AIRSTRIKE OPERATION AND TACTICS

An airstrike group is typically comprised of two or four aircrafts that fly in the formation shown in Figure 1. Aircraft number one is the strike group leader, followed by aircraft two and four being the second and third-in-command, respectively. Aircraft three is usually the least experienced, and a strike mission will usually be canceled when there is only an aircraft left. Within each group, aircrafts could be designated to be a decoy or bait to deceive the air defense systems, or designated to carry out the actual bombing of the targets. Decoy aircrafts tend to wander near the Lock-on Range of the air defense system (Figure 8) so as to lure the defense system to lock-on to them and opening up opportunity for the bombing aircraft to fly near the target to carry out the bombing

operation. Depending on the situation, those aircrafts that are designated to be decoys or bait could carry out the actual bombing when the opportunity arises.

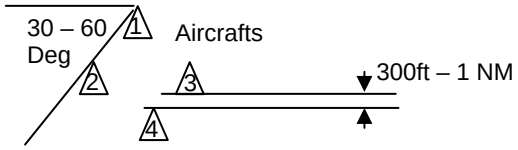


Figure 1 Formation of Airstrike Group

Aircrafts carrying out a bombing operation typically adopt the flight profile shown in Figure 2. During the initial ingress, aircrafts usually fly at an altitude above 10000ft to avoid Short Range Air Defense (SHORAD) from hitting them. When they are near to the target, they will drop to terrain-hugging altitude to avoid radar detection before climbing up to a certain altitude to loft the bomb towards the target.

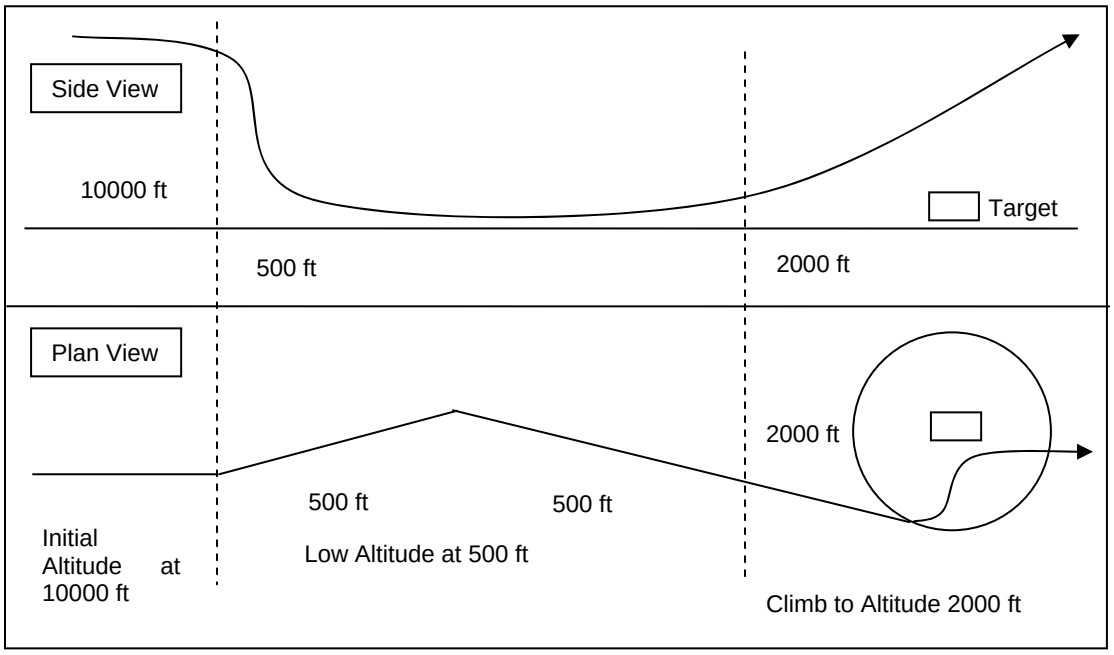


Figure 2 Flight Profile of Aircraft Carrying Out a Bombing Operation

In their approach towards the target area, aircrafts would usually identify gaps in the Air Defense Artillery (ADA) coverage based on prior intelligence that they have received. Once identified, the aircraft would exploit the gaps during their ingress as

shown in Figure 3. In the situation where no gap exists, the aircraft would then attempt to identify areas with the least coverage. These areas are the overlapping region between two ADAs as shown in Figure 4. As overlapping region is relatively shorter than other region, the aircraft could exploit the time required by the ADA to acquire and lock-on to them by transiting through the coverage area as quickly as possible.

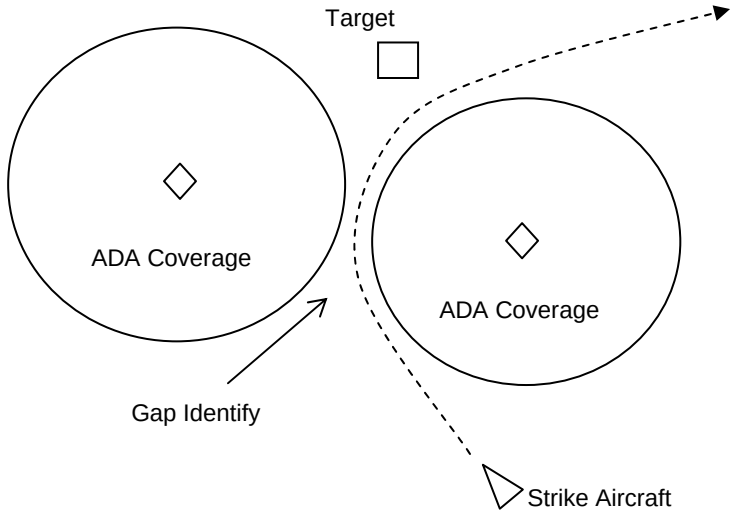


Figure 3 Aircraft Exploiting Gaps in ADA Coverage

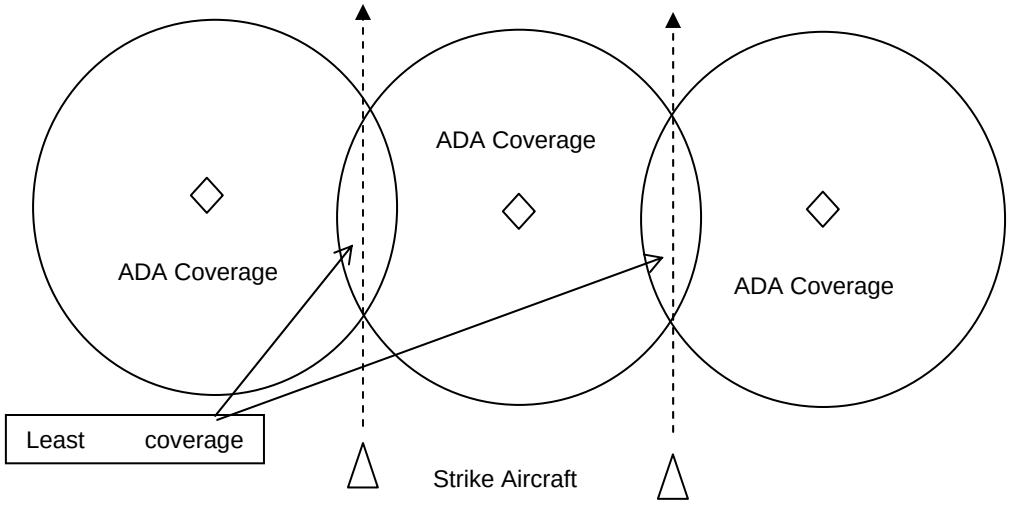


Figure 4 Areas with Least Coverage

When conducting strikes, airstrike groups would adopt different tactics in their attempt to overcome the ADA systems. These tactics include:

1. Saturation. In saturation tactic, the attacking aircrafts attempt to saturate the air defense sensor by flying many aircrafts into the defended area concurrently.

2. Deception and Prong Attack. In this tactic, the airstrike group is divided into two or three equal subgroups, and each subgroup will be assigned to fly into a target area using different axis (Figure 5). One of the subgroups is designated to be the decoy or bait group, while the others are tasked to carry out the bombing of the target. The decoy or bait group will usually make themselves prominently visible to the defender's radar.

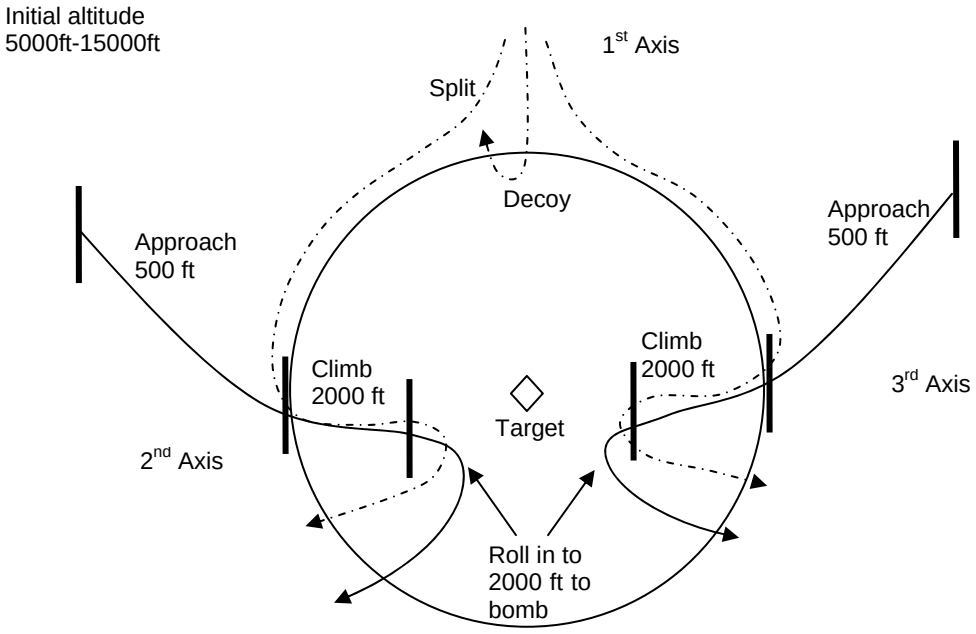


Figure 5 Deception Tactic

C. AIR DEFENSE PLANNING AND OPERATIONS

In a typical area defense plan, the area of interest covered by the integrated air defense system (IADS) is divided into Areas of Responsibility (AOR) as shown in Figure

6. Each air defense weapon (ADA) is assigned to provide protection to a primary AOR and a secondary AOR, which is adjacent to the primary. An ADA is usually assigned to either mark an incoming aircraft in its primary AOR or secondary AOR. For example, in Figure 6, primary AOR of ADA1 is AOR1 while its secondary AOR is AOR8.

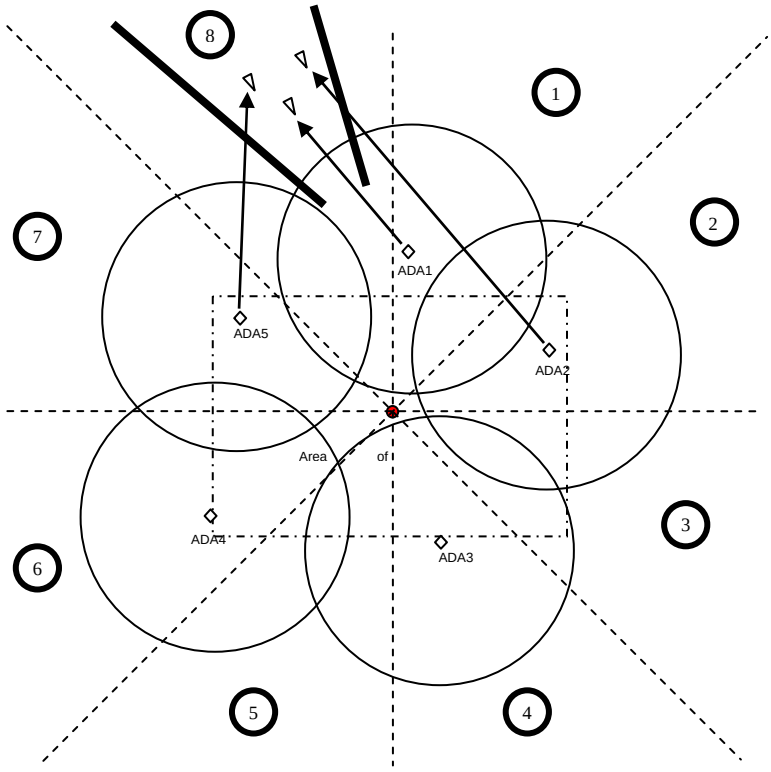


Figure 6 Example of an Air Defense Artillery Deployment

Each ADA's coverage is broken into four zones (Figure 7), namely, the Mark, Lock, Firing, and No-Firing zones. Incoming aircrafts are tracked in the Mark zone by surveillance radars that are usually located separately from the weapon site. In the Lock zone, the ADA's fire control radar locks onto the incoming aircraft and gets ready to fire. Depending on the type of ADA employed, there is a certain amount of time required for the ADA to break-lock on an aircraft and lock onto another aircraft. Once the aircraft flies into the Firing zone, the ADA will fire its weapon to eliminate the threat. ADA prioritizes their engagements based on the altitude of the target. An aircraft flying at low

altitude has a higher priority over those that are flying at higher altitude. ADA will not engage targets in its No-Firing zone as there is zero probability that its munitions could hit the target.

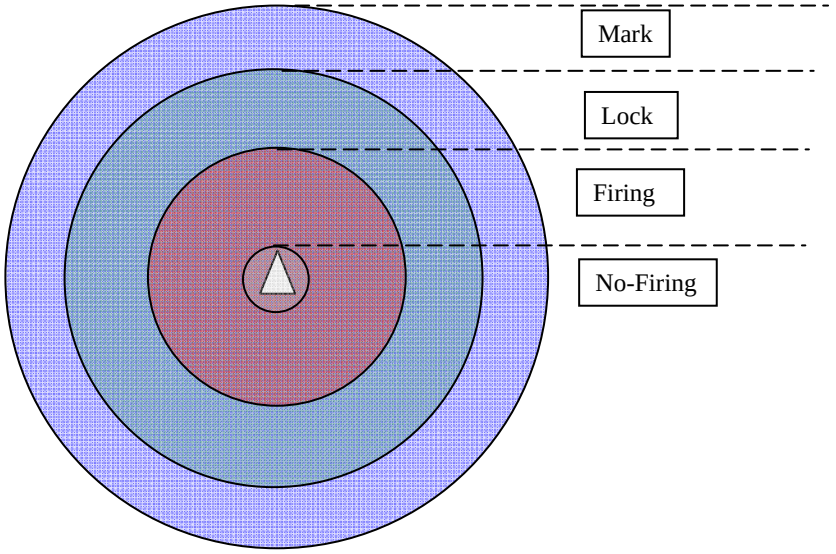


Figure 7 ADA Coverage Zones

The overall picture of an ADA operation is as shown in Figure 8. The “Heart” of an ADA is the sweet spot for firing the weapon as that range has the highest probability of killing the target. The figure also illustrates one of the common decoy tactics used by enemy aircrafts to tempt an ADA to lock-on to them.

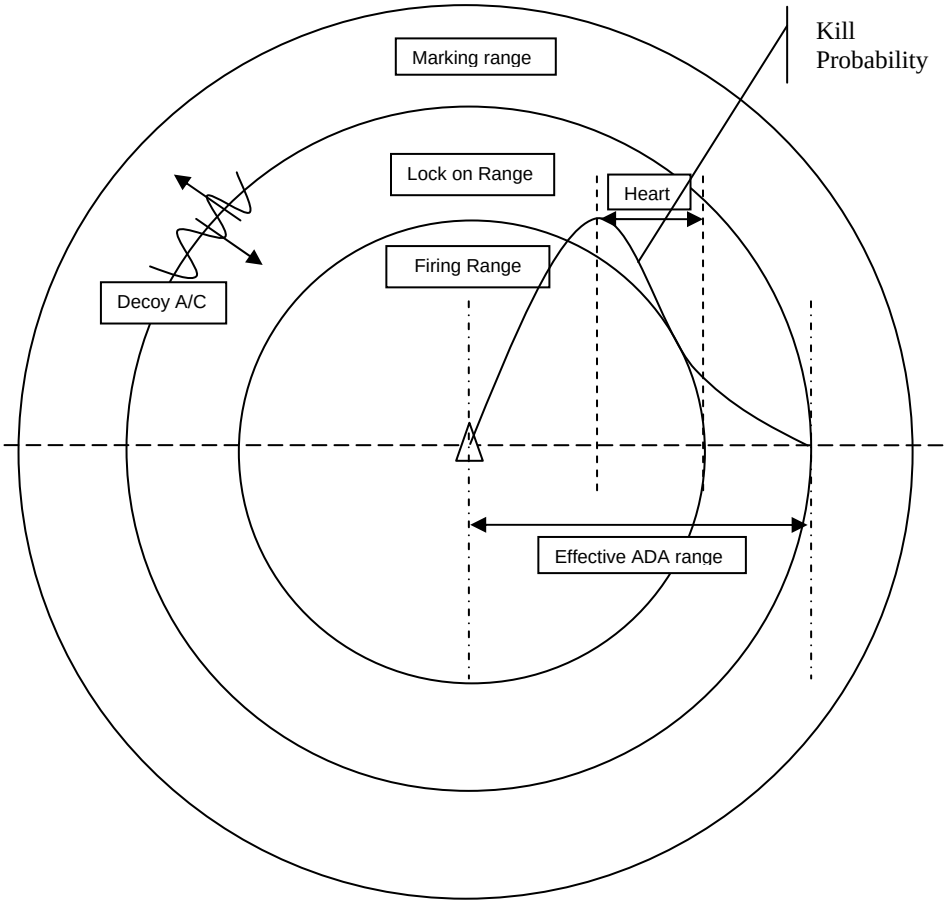


Figure 8 Overall Picture of an ADA Operations

D. THESIS OBJECTIVES AND SCOPE OF STUDY

This first objective of this thesis is to answer the question of whether Discrete Event Simulation can be used to model air defense plans, the interaction between air defense artillery and enemy aircraft. The second objective is to answer the question whether agents can generate quality attack plans that can be used for air defense plan validation. The final objective is to use the system developed to answer operational research questions, such as leakages for a certain scenario. Leakage is defined as a successful launch of munitions by adversary aircrafts. It does not take into account whether the adversary aircraft has egressed successfully.

The focus of this thesis is on Short Range Air Defense (SHORAD). It is assumed that most of the enemy aircraft would have been taken out by the medium to long range air defense with the remaining aircrafts to be tackled by the SHORADs. It further assumes that the aircrafts' objective is to breach the Bomb Release Line (BRL) and to drop the bombs that they are carrying. The egress route will not be considered and modeled in this thesis, as the Measure of Effectiveness (MOE) is the number of aircrafts that breached the BRL. In this model, it is further assumed that dumb bombs are used, which means that the aircrafts must fly to within certain range (BRL) to loft its bomb instead of firing longer range homing missiles. The SAM system modeled in this thesis is only capable of engaging one aircraft at a time. Lastly, this thesis will not model Suppression of Enemy Air Defenses (SEAD) scenarios as SEAD missions are complex operations that are beyond the scope of this thesis.

E. RELATED WORK

To the best knowledge of the authors, there has not been any prior work on the use of airstrike plans generated by MAS to evaluate or validate against air defense plans crafted by human planners. However, there is literature on the modeling of movers and sensors in DES. There is also much literature on creating agent behavior that could be applied in this thesis to generate intelligent air strike plans.

1. Related Work in DES Engine

In the world of simulation, high resolution models are often desired to provide detailed and more realistic simulation. However, as high resolution models are often more complex, they would take a much longer time to run as compared to low resolution models. Low resolution models, on the other hand, are simpler and faster to construct and run. They could complement high resolution models by identifying parameters of interest that could then be varied in high resolution models to simulate the detail effects of various settings of the parameters. Ahner, Jackson and Philips [1] have discussed the use of low resolution models for military analysis in their paper. As the objective of this

thesis is to generate attacker plans for evaluation of defender plans, it is desired to create a low resolution model to allow the defender plans to be tested with as many attacker plans as possible within a period of time.

The main elements in an airstrike/air defense scenario are: fighter aircrafts, SAM and AAG. Buss and Sanchez [2] have demonstrated that movements and sensors could be modeled in DES and implemented using Simkit. DES is preferred over traditional time-stepping approach due to the efficiency of DES in skipping through time-steps where no events are occurring, except for position updates. In DES, entity's position could be easily obtained by substituting the time of interest into the equation of motion, $x(t) = x_0 + (t - t_0)v$. If more detailed model of movement is required, the complex equation of motion could be used instead. This obliterates the need of using time-stepping for position updates as applications could now request for updates when required. In addition, Buss and Sanchez [2] have also illustrated the sensing process using DES. This allows the application to know precisely when is the enter/exit range events without checking at every time interval. The other major drawback about time step approach is when the delta time is not small enough, it is possible for objects to move over or right into obstacles, if the algorithm did not check for obstacles in every time step. For algorithms that do the checking, their performance could be affected quite significantly.

Buss and Ahner [3] have built a low-resolution DES model using DAFS (Dynamic Allocation of Fires and Sensors) as an example. In the DAFS example, they have demonstrated how sensing, movement and weapon effects could be modeled. This thesis embraced and extended the ideas presented in the paper to model different SAM ranges in a typical air defense setup.

2. Related Work in Multi-Agent System

In order to meet the objectives set out, the approach adopted was to explore the reviews of articles published by Artificial Intelligent game developers, to determine the best methods or concepts to build up the knowledge base, and to design an agent architecture to be used for the Plan validation application. In today's AAA

commercial game, whether the genre is strategy or first person shooter, all are guaranteed to use some form of agent-based model to simulate the adversary responses in one way or another to engage the user.

a. Hierarchical AI Architecture

Hierarchical AI [4] architecture describes an approach to implement artificial intelligent or agent-based models simulating as opponents who are capable of formulating strategy, rather than behaving predictably according to fixed sets of simple rules. These agent-based models can be describe loosely as Decision Making System (DMS) which can be a simple neural network, a collection of hard-coded rules, a set of fuzzy logic rules or a simple lookup table.

The aim is to make these artificial intelligent agent models to be as simple as possible such that it can take in a small set of data input and from these inputs generate a finite set of possible outputs based on its implementation.

The artificial intelligent agents are then classified into different level of DMS from a high overview level to the lowest implementation level. This is similar to how a general process strategic information and making decision based on the information thus presenting a chosen strategy to the commanders. The commanders, in turns, based on received tactical information and make tactical decision based on that information and the chosen strategy provided by the general, this process continues until the bottom level.

Each level of the DMS receives information directly from the level below it and then generalizes that information and presents the result to the level immediately above it. In return, it receives a set of objectives from the level above it and uses this set of objectives and the information from the lower level to compute a more precise set of objectives. Hence, the more precise set of objectives from that will become the input from above to the next lower level.

Therefore, the information filters up through the levels will be progressively more general, while the commands and objectives filter down through the levels will be more precise and detailed.

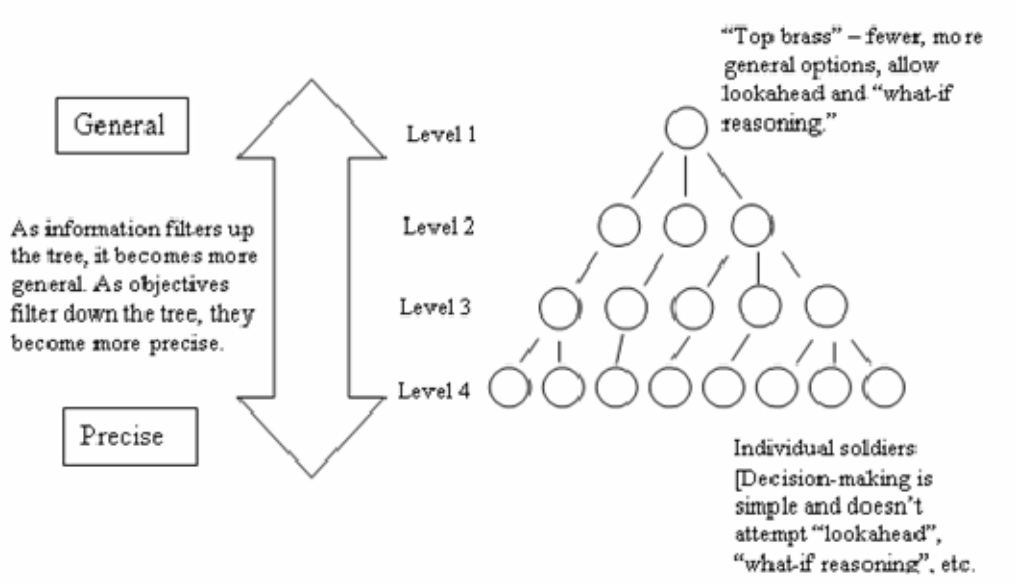


Figure 9 Hierarchical AI

The main advantage of this architecture is that it allows the top level of the hierarchy to formulate strategy based on information derived from its lower level. This will prevent the top level hierarchy from being overwhelmed by large numbers of decision-making possibilities occurring at lowest levels, which the agent-based model would have to consider if it possessed only information about individual soldiers. This will allow agent-based models at the top level to attempt "lookahead" or even to consider "what-if" scenarios.

In the related work, the computer game "Clash of Civilization" [5] adopted the hierarchical AI architecture. The genre of the game is a strategy civilization-building type, and the approach for designing the computer-based opponents is divided into several different levels in a hierarchical manner. At the top level of the hierarchy, there will be an AI agent processing long term strategy, while AI agents at the lowest level of the hierarchy will be attack staging.

b. Dynamic Procedural Position Evaluation Technique

One of the approaches to create a more responsive computer-based opponent (AI agent) is to make the AI agent, when selecting an action, evaluate its possible actions by taking into account of factors affect its current state in either a positive or a negative way and this type of technique is done in a real-time environment dynamically [6].

This technique is based on position evaluation functions very similar to a computer playing chess calculating the best possible series of moves. To evaluate a chess board position, a single value is computed from the weighted sum of several easily computed properties, such as: the number and weight of pieces, the mobility of each piece, the number of controlled board locations, pawn formations and king safety.

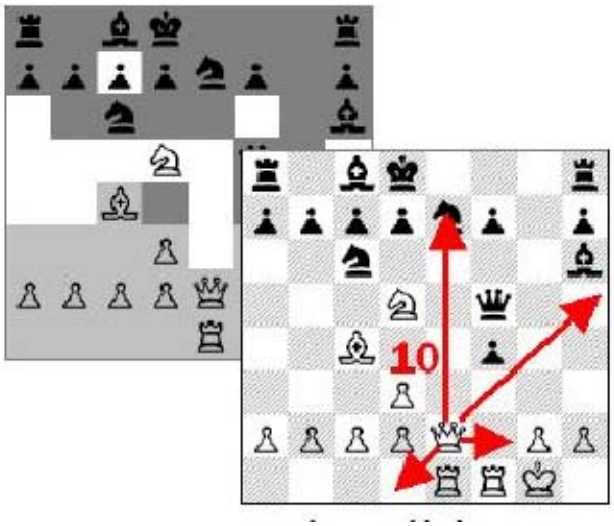


Figure 10 Positional Evaluation in Chess AI (From [6])

In the related work, the computer-controlled entity in the game Killzone [6], a first-person shooter genre uses dynamic procedural position evaluation technique. In the game, the computer-controlled entity pursues the goal that is most desirable in that situation. In order to pursue the goal, the entity executes a set of actions until the goal is achieved or becomes unattainable. A goal in the game can be combat goals such as the 'fire and maneuver' pattern: pick a destination, plan a path to destination, move along the

path, arrive and adopt a stance. Therefore, to pick a destination, the AI agent has to consider all positions within a radius around its current position or area of operation. From these, the AI agent eliminates positions that are already occupied by others and that are invalid. For each remaining position, the position evaluation function is executed to assign a score to the positions based on factors such as line-of-sight, closeness of target, cover and concealment. The higher scores suggest more attractive positions.

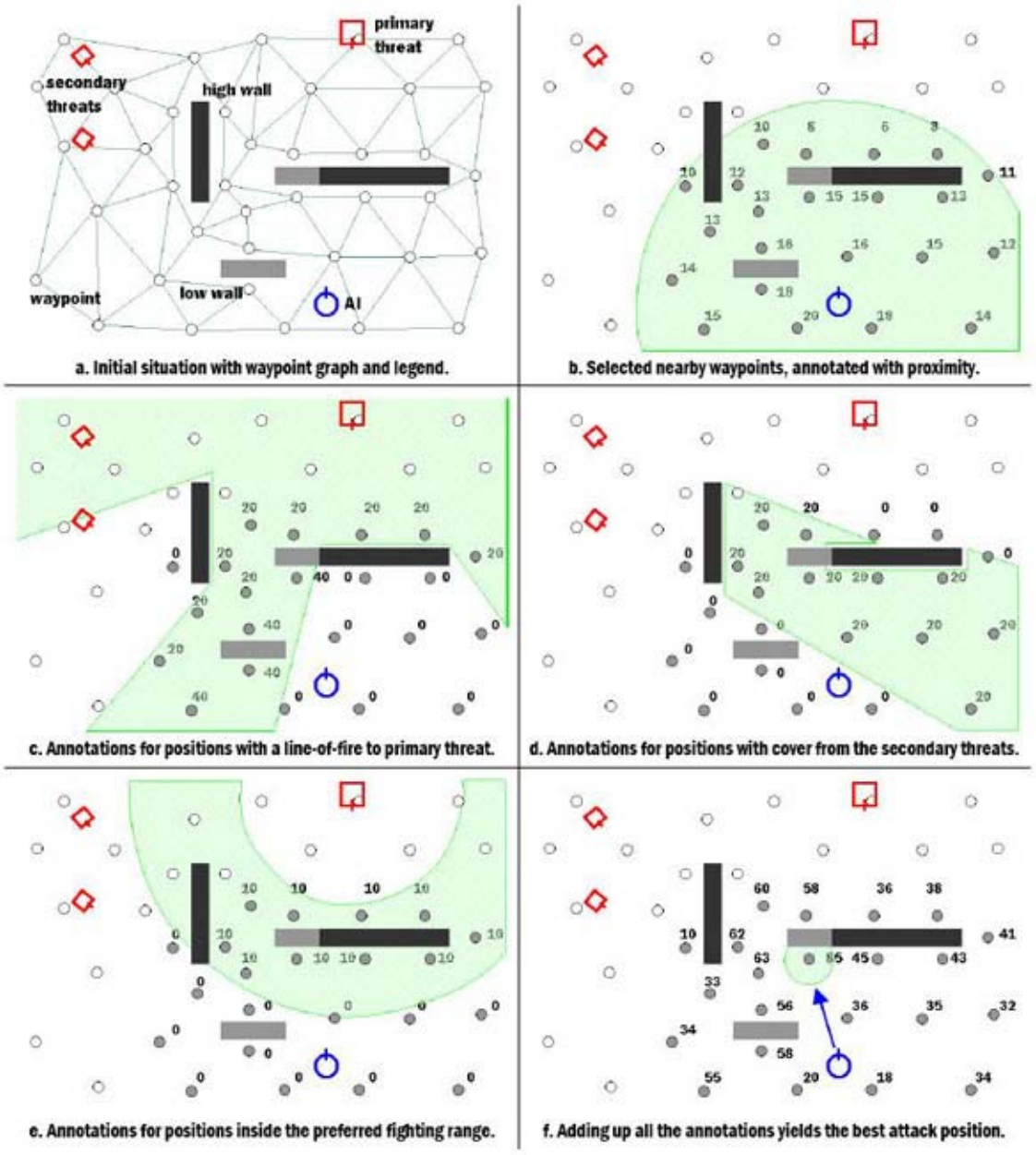


Figure 11 Tactical Position Selection (From [6])

c. Tactical Path finding

Path finding is used by computer games to calculate a path for travel from source to destination and it is usually the shortest path to destination. The algorithm that is used to calculate this is a standard A* algorithm.

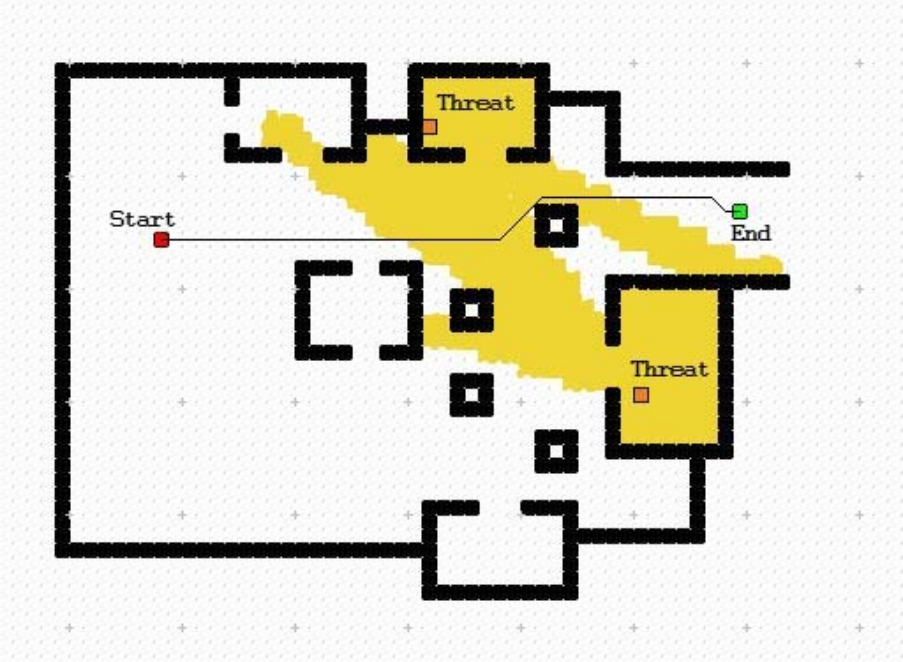


Figure 12 Standard A* Path Finding (Shortest Path)

Tactical path-finding [7] has the additional element of trying to reach the destination while the AI agent is taking into account being seen or shot at, and having to balance short travel times while avoiding hostile observation and fire.

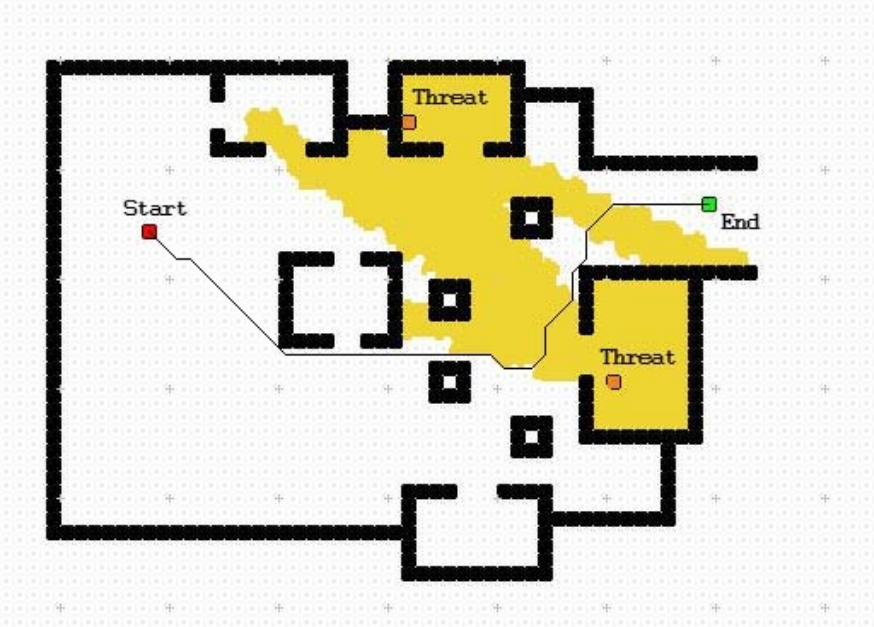


Figure 13 Tactical A* Path Finding taking into account of Protection and Concealment

In tactical path finding, the algorithm consists of the cost function, heuristic function, and a risk function. The cost function computes the exact cost of moving from one location to another location. These movement costs are computed from the distance traveled and speed allowed by the terrain. The heuristic function provides an estimate of the remaining costs to the destination. The risk function is used to calculate the additional cost for visiting locations in the enemy's line of sight, line of fire and distance of the threat to the visiting location. By adding the additional risk function as a cost factor to tactical path-finding, algorithms can take into account factors such as exposure time to enemy fire, locations that are exposed to the enemy, aiming duration, cover and concealment from enemy line of sight. Thus, the algorithm is able to calculate the most favorable path while balancing short travel time.

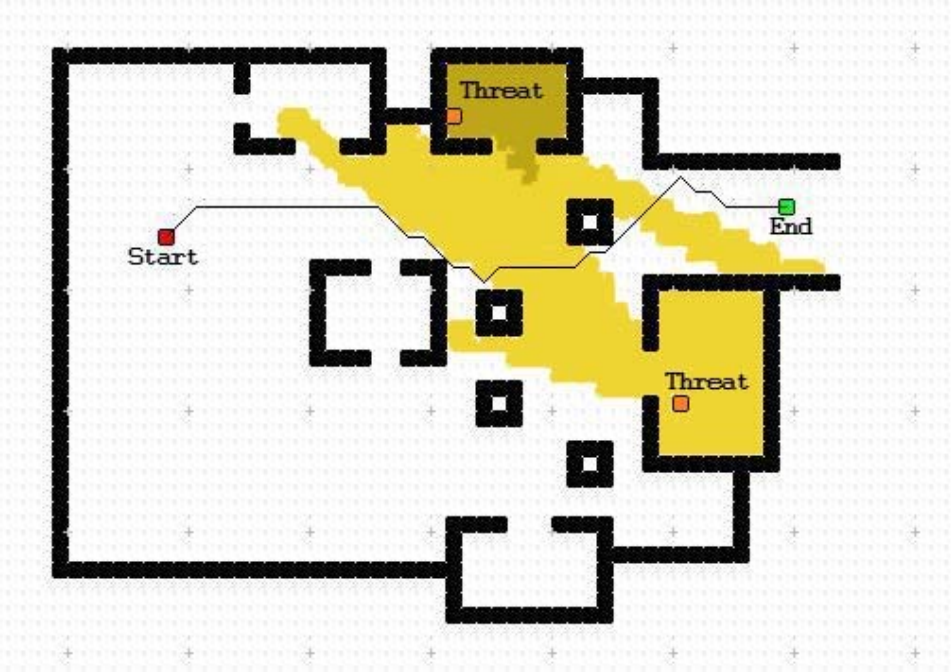
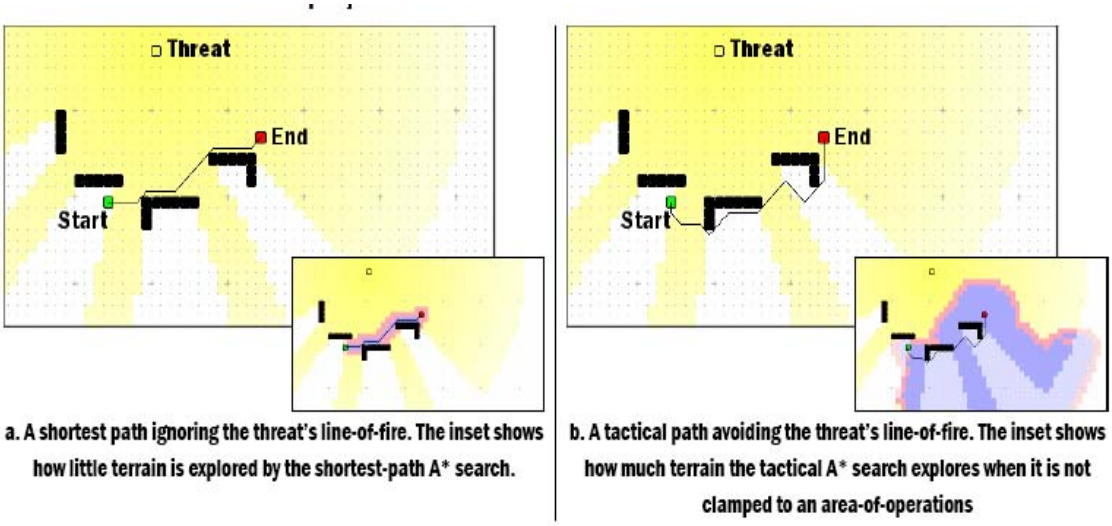


Figure 14 Tactical A* Path-finding taking into account of cover, concealment, threats and exposure duration to threat

In the related work, the computer game Killzone [6] also makes use of tactical path finding for its computer-controlled AI agent to take into account the threat of line of fire, so as to be able to detour and avoid enemy fire. The application also takes note of the increase in computational cost for path-finding in relation to AI agent having to operate in a big map area. Hence, areas of operation were assigned to each of the AI agents to control and limit where the AI agent fights and hides, as this will translate to a more efficient path finding.



a. A shortest path ignoring the threat's line-of-fire. The inset shows how little terrain is explored by the shortest-path A* search.

b. A tactical path avoiding the threat's line-of-fire. The inset shows how much terrain the tactical A* search explores when it is not clamped to an area-of-operations

Figure 15 Tactical Path in "Killzone" (From [6])

THIS PAGE INTENTIONALLY LEFT BLANK

II. METHODOLOGY

This chapter discusses the overall design of the system, the DES engine design and modeling, as well as the architecture and design of the agent-based strike plan generator.

A. OVERALL SYSTEM DESIGN

The plans validation system is comprised of a Discrete Event Simulation engine, an agent-based strike plan generator, a display interface, and other supporting components. The DES engine models the abstract behavior of air defense assets, air strike aircrafts, and the interaction between them.

The DES engine serves as a platform for validating air defense plans. The plans are evaluated by simulating the effects of agent generated strike plans against the defense plans. In other words, the DES engine forms the environment in which the aircraft agents operate. Environmental updates are communicated to the aircraft agents through User Datagram Protocol (UDP) messaging. Similarly, when the aircraft agents perform evasive maneuvers, they communicate their new waypoints back to the DES engine via UDP messages. Such communication allows real time interactions between the agents and the environment.

B. DES ENGINE DESIGN

The Discrete Event Simulation engine provides a platform for evaluating the air defense plan against the strike plan generated by the agent-based plan generator. It takes in the air defense plan crafted by human planners and the air strike plan generated by the agent-based generator. Both plans are stored in Extensible Markup Language (XML) files. Appendix C shows a sample strike plan while Appendix D shows a sample defense plan. The DES engine would then construct models of the components before starting the simulation.

The DES engine was designed based on the concept of Listener Event Graph Objects (LEGOs) framework [8], which allows a more complex model to be built in phases by linking smaller components together in a loosely coupled manner. The LEGO model of the DES engine is shown in Figure 67 of Appendix A. The DES engine was developed based on Simkit [9]. The main elements in the DES engine are: SAM system, Anti-Air guns and aircrafts. Other supporting components include: mediators, adjudicators and adapters. The model for the components will be discussed in more detail in the following subsections.

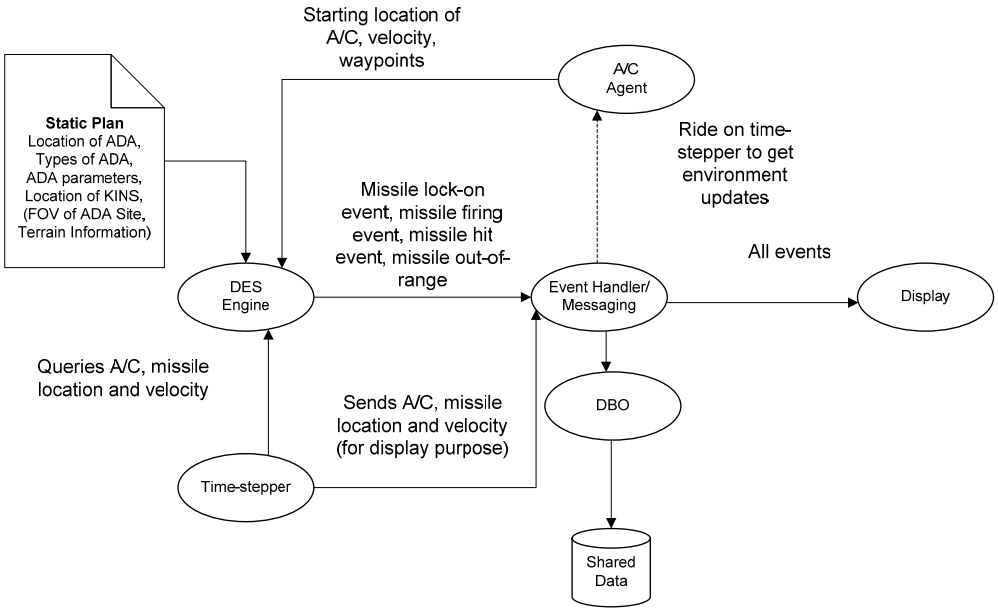


Figure 16 Overall System Design

1. Model of the SAM System

In the model, there is only one SAM system, which is the command and control unit of all the SAM sites in the DES engine. The event graph model for the SAM System is as shown in Figure 17.

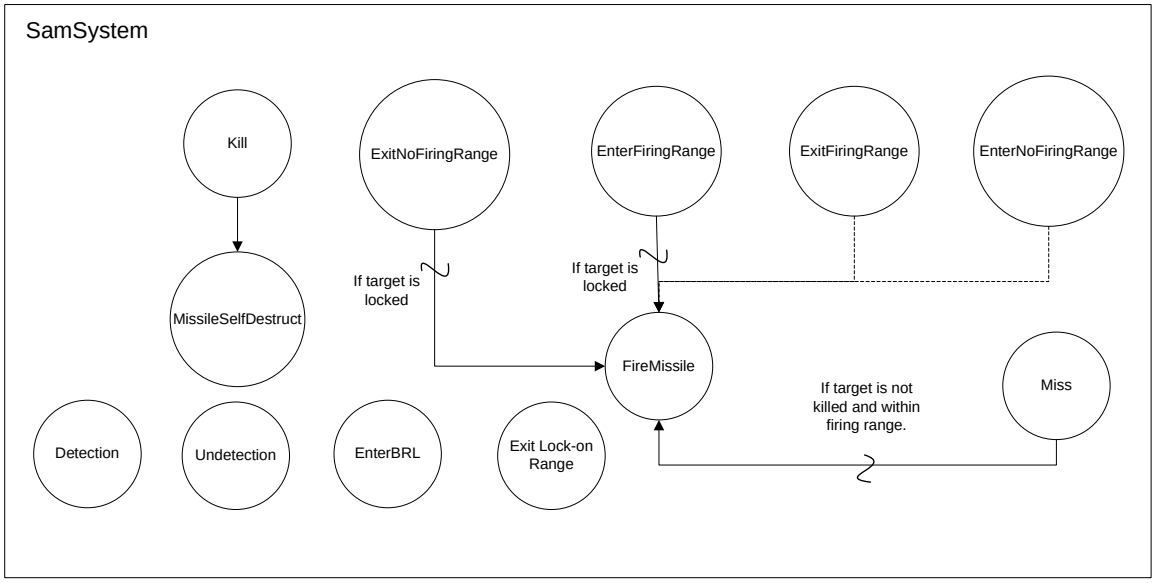


Figure 17 Event Graph Model of SAM System

The SAM system has the overall situation awareness of the defended area and performs target handover from a SAM site to the other whenever a target gets out-of-range of a SAM site. The system ensures that at any time, a target is engaged by only one SAM site. The firing doctrine adopted in the model is the SHOOT-LOOK-SHOOT, which means, firing a missile once locked-on, observe the result, and fire another missile if the earlier missile missed the target. The logical flow of the component is as shown in Figure 18. The oblong circles in represent the “do” methods. Detection and Undetection events are scheduled by the SAM Mediator after a detection/undetection delay. The Miss and Kill events are scheduled by both SAM Adjudicator and the Gun Adjudicator depending on the adjudication’s outcome. Details of the adjudication will be covered in subsequent subsections. The EnterBRL event is scheduled by the BRL Mediator, while the ExitLockOnRangeEvent is scheduled by the Exit Lock On Range Adapter.

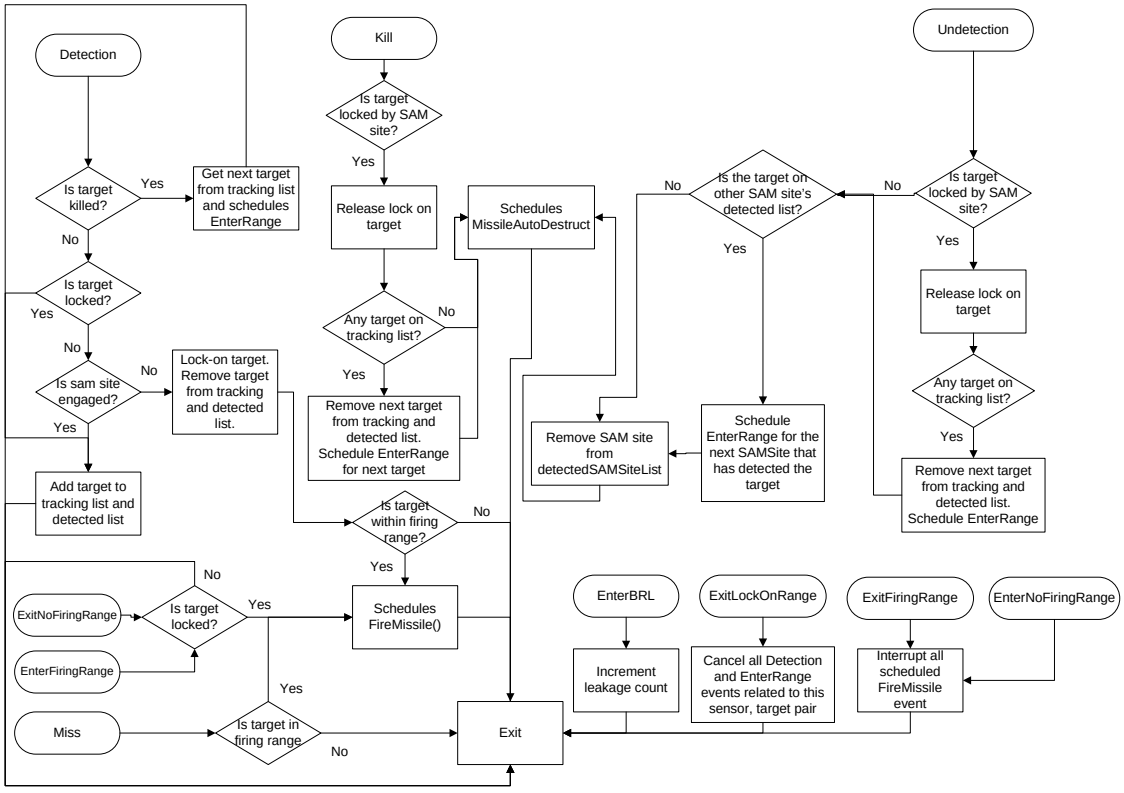


Figure 18 Flowchart for SAM System Component

As mentioned earlier, a SAM site is only capable of locking onto one target at a time, while the other targets that come into the lock-on range of the SAM sensor are tracked. When the locked-on target is destroyed, or the target exits the lock-on range, a new target is selected from the tracked list based on a first-come-first-serve policy. The missile that is fired will also be scheduled to be destructed once the target is destroyed or becomes unlocked.

2. Model of a SAM Site

Each SAM site comprises of a sensor and a missile launcher. In the model, a SAM site is only capable of locking and engaging one target at a time. A simple target selection algorithm is implemented based on the first-come-first-serve principle. Thus, when more than one target enters the sensor range, the first target will be locked-on, while subsequent targets will be put on the watched list. Whenever the first target is

destroyed or gets out-of-range, the next target on the list will be acquired and locked. The model has assumed perfect command and control, where a target is always handed over to the next SAM site that has the target in its sensor range.

The event graph model for the SAM site is shown in Figure 19. The missile launcher is loaded with a configurable number of missiles. The number of missiles in the launcher is decremented whenever a missile is fired. When all the missiles in the launcher are expended, a reload time will be incurred to reload the missile launcher. The SAM site listens to the SAM system's event. If the SAM system schedules a FireMissile event and the missile launcher is not empty, a missile is launched through the Chase event, which then invokes methods in the DAFSInterceptMoverManager to control the movement of the missile. The range of the missile is checked at regular intervals, and when it gets out-of-range, the missile will be destructed.

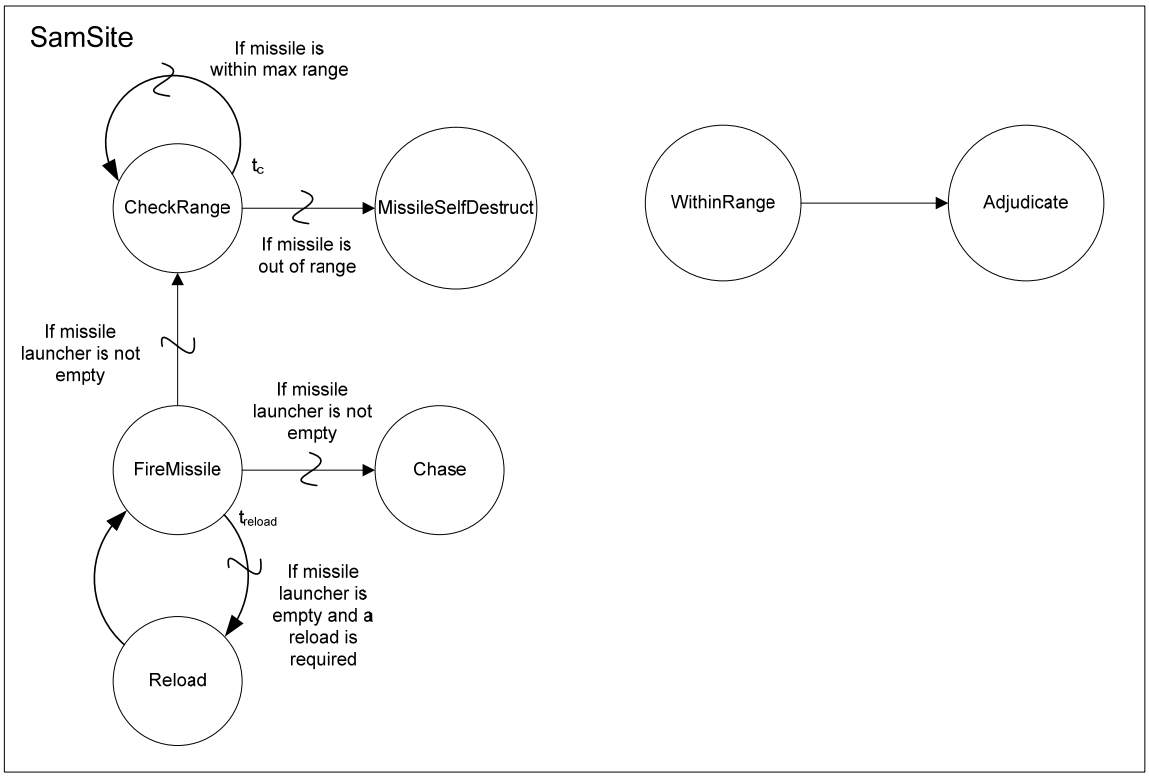


Figure 19 Event Graph Model of SAM Site

The SAM site component listens to the InterceptMoverManager component. When the InterceptMoverManager checks that the missile (mover) is within a configurable range (proximity) of the target, it schedules the WithinRange event, which is then pickup by the SAM site. The SAM site in turns schedules the Adjudicate event to the SamAdjudicator component. The proximity is used here to model SAM equipped with a proximity fuse.

3. Modeling a SAM Sensor

The use of DES for simulation of sensors and movers is not new. Buss and Sanchez [2] detail how movement and sensing can be modeled using DES. While the Gun sensor is modeled after the concept presented, the SAM sensor is developed by extending the concept for sensing mover at multiple ranges.

A typical SAM sensor is capable of sensing targets that enter or exit its sensor range (lock-on range), weapon range (firing range) and the weapon's no-firing zone. In the SAM sensor model, three different sensors are used to model the various sensor ranges with their center aligned to the same static mover, which simulates the platform that the SAM sensor is installed physically. Figure 20 depicts how the SAM sensor is modeled, with the blue triangle indicating the location of the common platform.

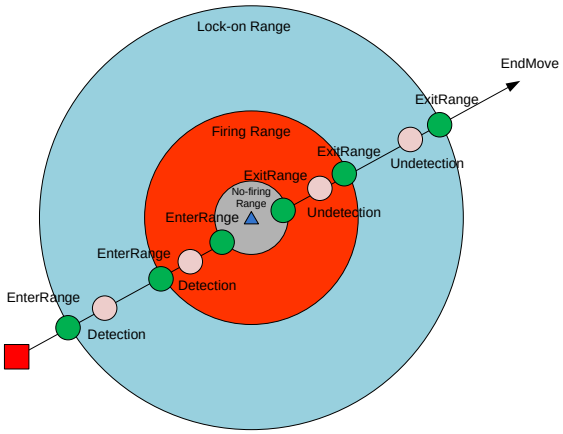


Figure 20 Various SAM Ranges

When an aircraft enters the “Lock-on Range” sensor, it will be detected and locked by the SAM system after a fixed amount of delay (SAM reaction time). Upon entering the “Firing Range”, a missile will be fired at the target after a certain amount of delay (SAM engagement time). When a target enters the “No-firing Range”, the SAM will not engage the target as there is too little time for the missile to be launched and catch up with the target.

The same “EnterRange” event triggered by the three sensors creates an issue as all “EnterRange” events are to be handled by the same mediator. The mediator is unable to differentiate which sensor initiated the event, since they have the same event name. Buss [10] proposes a simple but elegant way of overcoming such situation with the use of an “adapter” class. The “adapter” class listens to an event and triggers a new event. The same mechanism is used to overcome this problem as shown in Figure 21. The adapter works by adapting the same “EnterRange” event produce by the three sensors into specific events: EnterLockOnRange, EnterNoFiringRange and EnterFiringRange.

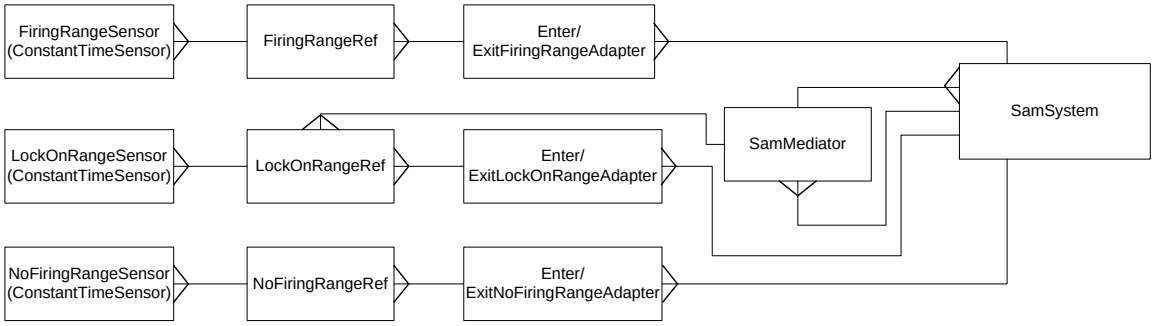


Figure 21 Use of Adapters to Differentiate EnterRange Events

4. Modeling SAM Trajectory

A simple path mover manager, which moves the SAM to an extrapolated interception point, is insufficient to model the trajectory of a SAM realistically, as a missile needs to respond to its target’s maneuver by changing its own trajectory. The SAM model is built based on the concept of the intercept mover manager [3]. The strength of an intercept mover manager is that it tracks the location and velocity of its

target at regular time interval, re-computes the projected interception point and moves the missile to towards the revised interception point. The intercept mover manager continues to track and re-project new interception point, until the mover is within certain proximity of the target.

The event graph model for the InterceptMoverManager is shown in Figure 22. In addition to EndMove event, the UpdateIntercept event is also triggered by the change in the State property of the missile (mover). UpdateIntercept schedules the WithinRange event if the missile is within the specified proximity of the target. The WithinRange event is handled by SAM site as detailed in an earlier subsection. If the missile is not within the proximity of the missile, UpdateIntercept re-computes the new intercept point, based on the current location and velocity of the missile. If a valid intercept point is computed, the missile will be moved to the new intercept point. If there is no valid intercept point, the CantPursue event will be scheduled, since the missile is unable to intercept the target.

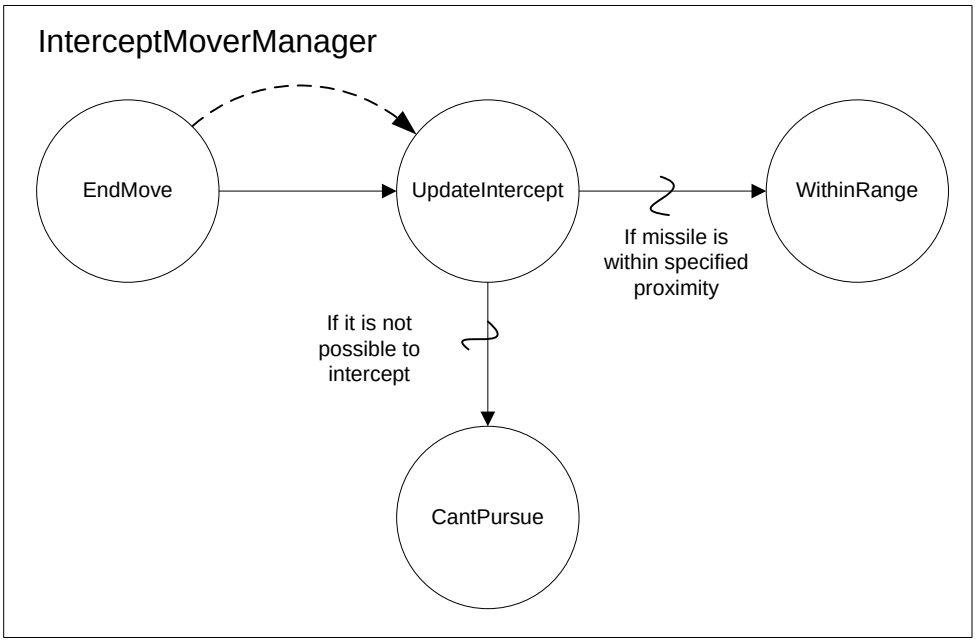


Figure 22 Event Graph Model of Intercept Mover Manager

In the event that a SAM flies out of its maximum range, the DES engine simulates a loss of command link and the SAM will be self destructed after a certain delay. This is a common feature available in current SAM systems.

5. SAM Probability of Kill

When the Adjudicate event occurs, the SamAdjudicator determines if a target is killed based on a probability of kill (Pk) value. In the model, a triangular Pk contour, as shown in Figure 23, is used instead of using a single-valued kill probability, to create a more realistic model. This Pk contour could be easily substituted with any other more accurate Pk contours in the future when they become available. Based on SME's input, the highest SSKP is usually achieved at around 70% of the maximum weapon range. As such, the vendor supplied Pk becomes the height of the triangle in the Pk contour.

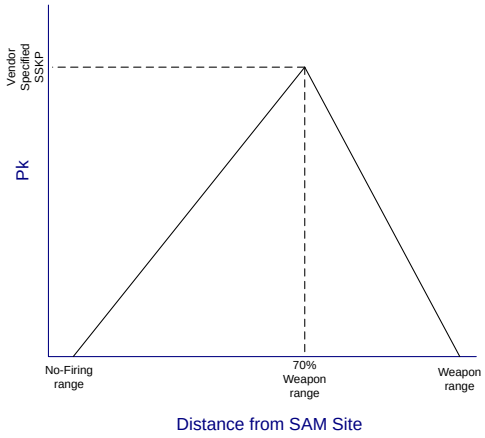


Figure 23 Triangular SAM Pk Contour

During the simulation, a target is considered as hit when the SAM is within certain proximity of the target. A uniform(0, 1) random number is generated using the Mersenne Twister (MT) random number generator. MT was chosen due to its long period of $2^{19937} - 1$ and a low working memory of 624 words [11]. The random number is then compared against the Pk value corresponding to the range of the target. Any number smaller than the Pk value, will be considered as a kill. Otherwise, it is considered as a miss and another missile will be scheduled to engage the target again.

6. Modeling the Anti-Air Gun

According to the Field Manual 44-43 [12], a high volume of fire is desired to increase the probability of kill when engaging aerial targets with guns. Thus, unlike the

SAM system, there is no restriction on the number of anti-air guns allowed to engage a target simultaneously. The guns will engage any targets that get within the weapon range of the gun. As in the SAM system's case, the gun will engage its targets on a first-come-first-serve basis. Subsequent targets that enter the weapon range of a gun will be kept on a list. When the first target is destroyed, the gun will engage the next target on the list sequentially.

As in the case of SAM system, the anti-air gun also adopts the SHOOT-LOOK-SHOOT policy. When a target enters the weapon range of a gun, the gun will release a burst of 50 to 100 rounds of munitions at the target. If the target is killed, the gun will move on to engage other targets in its target list, otherwise it will release another burst. The number of rounds to be released in each burst is based on a uniform(0, 1) generated by the MT random number generator and scaled to a number between range of 50 to 100. For each round of munitions released, the ammunition count is decreased by one. When the munitions run out, a delay is incurred for the reload event.

The event graphs for Gun Site and Gun are shown in Figure 24. The GunEngage event is scheduled in the GunSite component when a target enters the gun's range. The event is also scheduled when a target is killed or when a target gets out-of-range and there are more targets tracked by the GunSite. In the Gun component, the GunEngage event is scheduled recursively until a target is killed. A Reload event is scheduled when the gun is out of ammunitions. As each GunEngage event simulates a burst of rounds, an adjudicate event is scheduled to determine the munitions' effect on the target. The GunAdjudicator picks up the scheduled Adjudicate event and determine kill or miss.

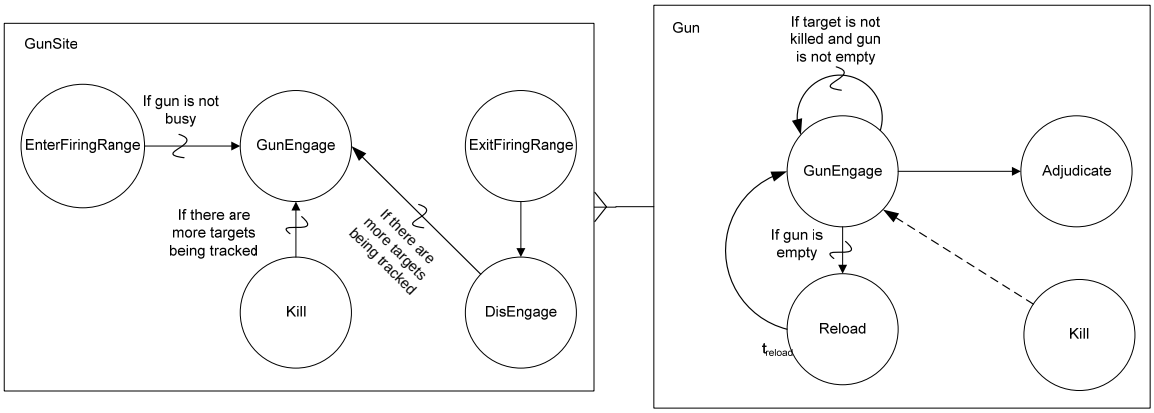


Figure 24 Event Graphs for Gun Site and Gun

As the anti-air gun in our model is not associated with any fire control radar, their target detection event could be modeled simply using a constant time sensor [2].

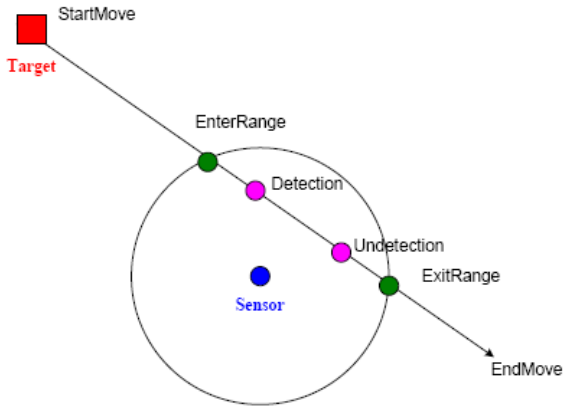


Figure 25 Anti-Air Gun's Sensor

The trajectory and the probability of kill for each round of gun munitions are not modeled explicitly in the resolution of this model. Instead, the probability of kill for each burst of bullets is used to adjudicate the effects of the munitions on the target. In the gun's Pk contour shown in Figure 26, it is assumed that the number of rounds released at the target directly affects the Pk. When the number of rounds released exceeds 50, the Pk will be capped at the vendor's specified SSKP. As in the SAM system, the Pk contour could be substituted with more accurate ones when they become available.

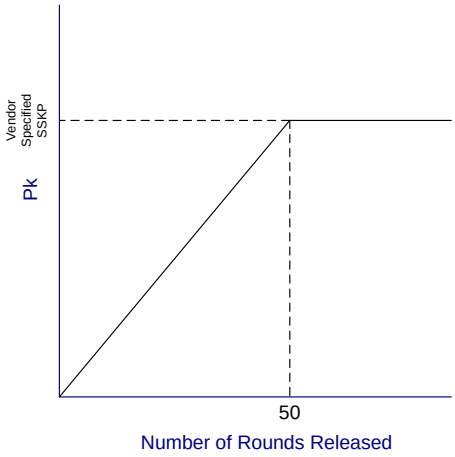


Figure 26 Pk Contour of Anti-Air Gun

7. Modeling the Strike Aircraft

Strike aircrafts are simply modeled using Uniform Linear Movers which are controlled by Path Mover Managers. In the current DES model, the aircrafts are able to vary their speed at each waypoint. However, acceleration and deceleration are not modeled.

8. Modeling the Target Site

In this model, the key Measure of Effectiveness (MOE) is the number of leakages through the Bomb Release Line (BRL). As the MOE is only concerned about number of leakages, a simple Cookie Cutter sensor is sufficient for tracking number of aircrafts (movers) that entered the sensor's range. It is assumed that any aircrafts that entered the BRL is bound to launch their bomb successfully.

C. AGENT-BASED STRIKE PLAN GENERATION

The second portion to this system is the Agent-Based model; this model will be used to support the validation of the Air Defense plan thru the use of the Discrete Event Simulator. In this section, we will describe the architecture and algorithms that were designed to meet the requirements illustrated in the Air Strike Operation section.

1. Agent-Based Model Architecture

The eventual goals of agent based model are to generate appropriate strike plans for the agents representing strike aircraft, and implementation of a behavior model for the agents in the simulation environment.

The architecture of the Agent-Based Model is based on a hierarchical decision making process similar to the Hierarchical AI approach [5]. Instead of having an overall agent in charge of making plans, deciding where to strike, and determining how many aircraft formations should be created, the idea is to breakdown the decision-making process into levels.

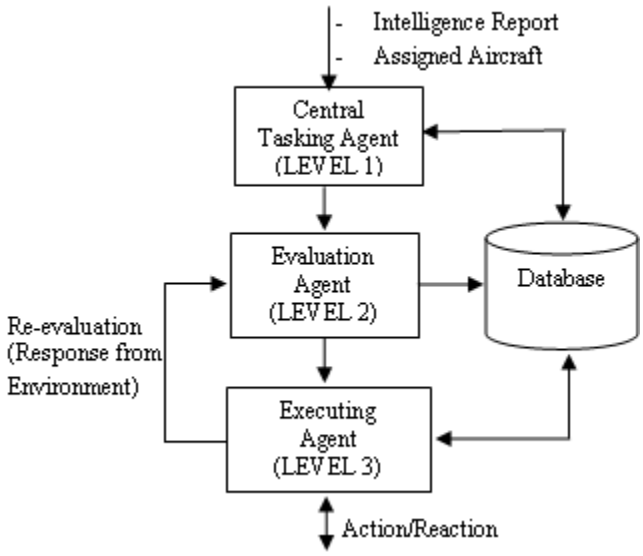


Figure 27 Agent-Based Model Architecture

This is roughly analogous to the chain of command in an army where the general's overall goal is to provide a strategic plan to guide his commander, and the commander makes tactical decisions based on the strategic plan, and so on down to privates who are executing the plan. In the figure above, the agent-based model architecture consists of three levels of decision-making processes. At the highest level is the Central Tasking Agent, which is responsible for generating the number of participating air formations, assigning area of operations and targets to the participating formations. The assignment is based on intelligence information gathered on the target's

air defense and the number of aircrafts that are assigned to the strike operation. This is akin to Level one of the decision making hierarchy in Figure 9. The Evaluation Agent is at Level two of the hierarchy, which receives information on air formations, assigned area of operation and target from Level one agent. It will generate the participating aircrafts in the air formation and also generate a suitable course of action for the air formation. The Executing Agent is at Level three of the decision-making hierarchy, which is the lowest level in the agent architecture. The Executing agent is like the foot soldier in the army, receive marching orders from the Evaluation agent, such as target objective and approach to the objective. This agent will receive real-time information from the simulation environment and based on the information received, it will act on it and at the same time relay the information back to Level two of the decision-making process so that the decision making agent at Level two can re-evaluate the course of action.

a. Agent Descriptions

In the context of our paper for Air Defense plan validation, the generation of Air strikes and behavior model for the air threats, the Central Tasking Agent will receive inputs such as intelligence reports, number of aircrafts available to be used etc. The Central Tasking Agent will then make use of the input data provided to generate an approach vector to the strike zone depending on the tactics to be adopted. It will also be responsible for creating the number of strike formations and assigning the area of operation for each of the strike formations.

Evaluation Agents will be assigned to each of the strike formation created by the Central Tasking Agent. The agent will be receiving data input from the Central Tasking Agent such as approach vectors, number of assigned aircrafts for the particular strike formation and area of operation. It will then generate an appropriate route to the strike zone using the assigned approach vector. The evaluation agent also creates the strike aircrafts that belong to its formation.

Executing Agents are assigned to each of the strike aircraft created by the Evaluation Agent. The Executing agent will have a behavior model based on the

attributes of the strike aircraft assigned by the Evaluation Agent. Information perceived by the Executing Agent will be consolidated into a central database to be used by the Evaluation Agent.

The architecture tasking of agents is illustrated by the following diagram:

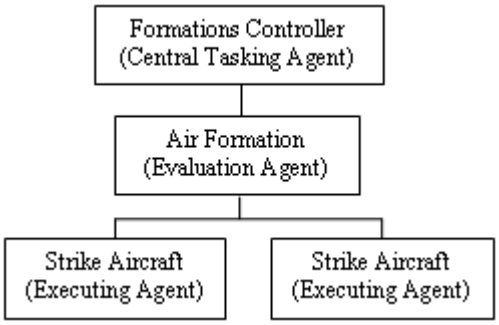


Figure 28 Agents Tasking

2. Approaches for Planning and Control

In the previous section, the agent architecture model described a hierarchical approach for making decisions. The central tasking decides how best to conduct a strike into the strike area, the evaluation agents plan the actual movement of strike aircrafts with this information, and the executing agents execute the movement plan accordingly. In the following paragraph, the algorithms that were used by these agents will be discussed.

When determining how to best approach the strike area, a proposed technique similar to position evaluation function described in KillZone AI [6] was used. Position evaluation functions are well known in computer chess, where the AI generates possible board positions and evaluates these board positions to select the strongest series of moves.

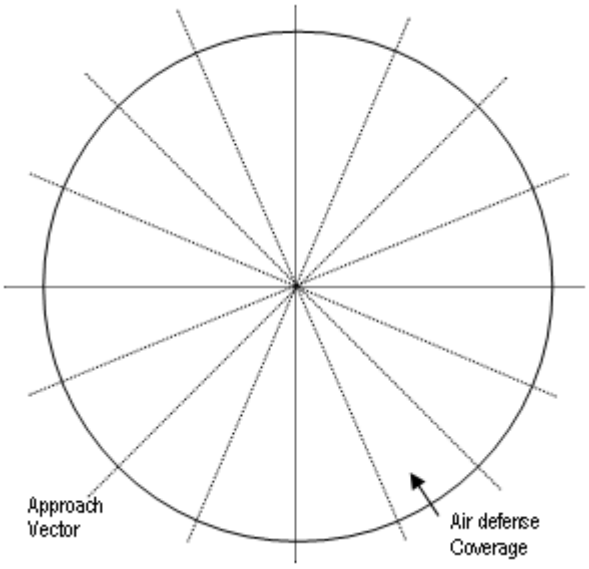


Figure 29 Position Evaluation for Approach Vectors

In this technique, there is a need to find out the best approach vectors to the strike area. The various factors that have to be considered are based on the cognitive task analysis that was conducted with a subject matter expert. These factors are the air defense coverage, overlapped air defense coverage, exposure time to the air defenses, exposed distance to air defense before reaching the strike area and the speed of the aircraft. The number of approach vectors also depends on the various types of tactics to use. To generate a suitable approach vector, the strike area has to be determined first. From the strike area, straight lines are generated for every 10 degrees. The eventual result will look like the spokes in a wheel. For every spoke line, we first determine the total exposed distance to the air defense, and normalize the value which will be used as a score. The exposed distance calculation is based on the line-intersection of the air defense coverage. The total exposed distance of a single spoke line is obtained by adding the exposed distance that a spoke line intersected with individual air defense coverage:

$$\text{Expose distance of a single spoke} = \sum \text{Distance of spoke intersecting air defense coverage}$$

The exposed distance is normalized by the following formula:

$$\text{Normalized Exposed distance} = \frac{\text{Exposed distance of a single spoke}}{\sum \text{Exposed distance of spokes}}$$

Secondly, there is also a need to determine the exposure time to the air defense coverage for each of the spoke lines. This is due to the fact that the exposed distance alone is not good enough to determine the best approach for overlapped air defense coverage as this also depends on the exposure time over this composite air defenses coverage. The strike aircraft is required to cross this exposed distance as quickly as possible. Therefore, the exposure time calculation is based on the following formula:

$$\text{Exposure Time} = \frac{\text{Exposed distance of the composite air defenses coverage}}{\text{Speed of the strike aircraft}}$$

The exposed time over the composite air defenses coverage has to be normalized and this is obtained by:

$$\text{Normalized Exposure Time} = \frac{\text{Exposure time of a single spoke}}{\sum \text{Exposure time of the spokes}}$$

The normalization of these 2 scores is to create a unify metric for selecting the best approach. These 2 normalized scores are then added together to represent the weight of the approach vector and the best approach will be the vector that has the highest scores.

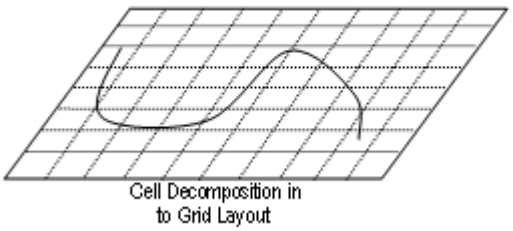


Figure 30 Cell Decomposition Search space

In the approach for movement planning, the cell decomposition approach is chosen as described in Movement Behavior for Soldier Agents on a Virtual Battlefield [13]. The idea is to represent free space and air defense coverage as a grid of small uniform cells

that are square in shape. Although the cell cannot represent the shape of the air defense coverage exactly, it is possible to vary the size of the cell to either increase or decrease the details of the representation. The size of the cell is always inversely proportional to the detailed level of the representation. The movement planning on the grid is by searching through the cells. Once the area of operation is represented in a grid, which can also be know as the threat map, the A* algorithm is typically used to control the search from start to destination, with the straight line distance to the destination as a heuristic function. A simple A* search algorithm is not used; instead a technique described as Tactical Path-Finding with A* [7]. This search algorithm still follows the generic function of the A* search algorithm where G_x is the cost function, H_x is the heuristic function and F_x is the sum of the cost functions and the heuristic given by:

$$F_x = G_x + H_x$$

In this technique, additional considerations are factored in the cost function, G_x, of the algorithm, which include the exposure to air defenses. The exposure cost are based on the type of air defense unit covering the area, for overlapping coverage of two or more air defense unit, the cost are sum together. The heuristic, an estimate of the minimum distance from start to end, use is Euclidean distance function which is an application of Pythagorean Theorem between start point, (S_x, S_y) and end point, (E_x, E_y) is given as:

$$\text{Euclidean distance} = \sqrt{(S_x - E_x)^2 + (S_y - E_y)^2}$$

Hence, the costing structure of the air defense type that is deemed to be suitable is as follows:

Cost Structure	
SAM	10
GUN	5
MOVEMENT COST	1

Table 1 Cost Structure for path planning algorithm

Lastly, behavior mechanisms for responding to state changes in the environment are added to the individual agent. This behavior mechanism includes a set of actions and a steering behavior. The typical states convey back to the agent from the environment with reference to the air defense system are “Lock-on”, “Lock-off”, “Incoming missile” and “Gun firing”. The current actions implemented, which can be taken by the agent include evasive process and strike process in response to the state of the environment. Each of the agents keeps track of its own current action and process an action if the current action is not suitable of the change of state received from simulation environment. The current action of the agent can be the following “Lock-on Action”, “Lock-lost Action” and “Evasive Action” actions.

The process action of “Evasive Action” is under taken when the agent received a state change message from “Lock-on” state to “Incoming missile” state from the simulation environment and the action of the agent is not ‘Evasive Action”. The agent will initiate an evasive process when its action is “Evasive Action”, and will generate a series of waypoints out of the strike area based on its current heading, and the direction away from the target area, the waypoints are then send to the simulator which will reflect the agent steering behavior.

The process action of “Lock-lost Action” occurs when the agent received a state change message from “Incoming missile” state to “Lock-off” state from the simulation environment and the agent action is not “Lock-on Action” and “Evasive Action”. The agent will initiate a strike process when its action is “Lock-lost Action” and will generate waypoints back to the target area. The waypoints will be sent to the simulator which will reflect the agent steering behavior.

THIS PAGE INTENTIONALLY LEFT BLANK

III. TESTING AND EVALUATION

This chapter describes the verification and validation process conducted for the system developed. While verification could easily be done based on the flowcharts and logic diagrams drawn during system design, validation could only be done through face validation by the SME.

A. VERIFICATION OF DES ENGINE

Using the LEGO framework allows various building blocks (Figure 67) of the DES engine to be tested independently before linking them up for a complete test. As the sensor classes, sensor-target referees, mediators, and mover managers are instantiated based on Simkit classes, it is assumed that they have been reasonably verified and they will not be tested again. The DES engine verification will focus on the main components that were developed in this thesis, such as, the SAM system and its related components and the gun site and its related components.

1. Verification of the SAM and Anti-Air Gun Adjudicators

In order to verify the behavior of the SAM Adjudicator, a test application was written to vary the target's range from the SAM site. The resulting kill probability (Pk) was collected and plotted against the range as shown in Figure 31. It can be seen from the figure that the Pk contour resembles that of the theoretical contour shown in Figure 23. Thus, the SAM Adjudicator has been verified to be behaving correctly.

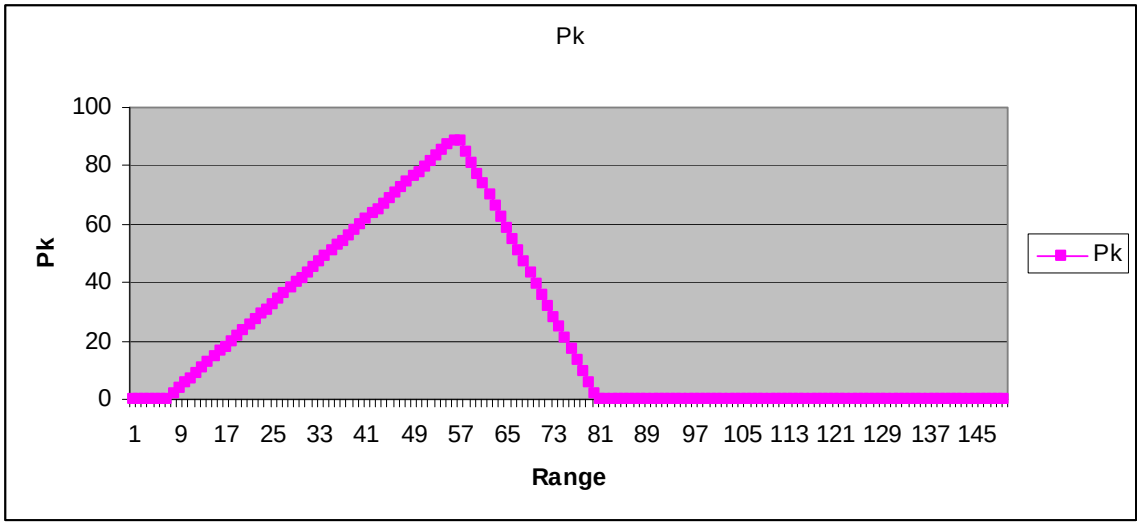


Figure 31 Pk Values Computed by SAM Adjudicator

The gun adjudicator was verified using similar methods. A test application was coded to vary the number of rounds fired to plot the Pk contour for the gun, as shown in Figure 32. As the Pk contour for the anti-air gun resembles the theoretical contour shown in Figure 26, the anti-air gun adjudicator is assumed to be behaving correctly.

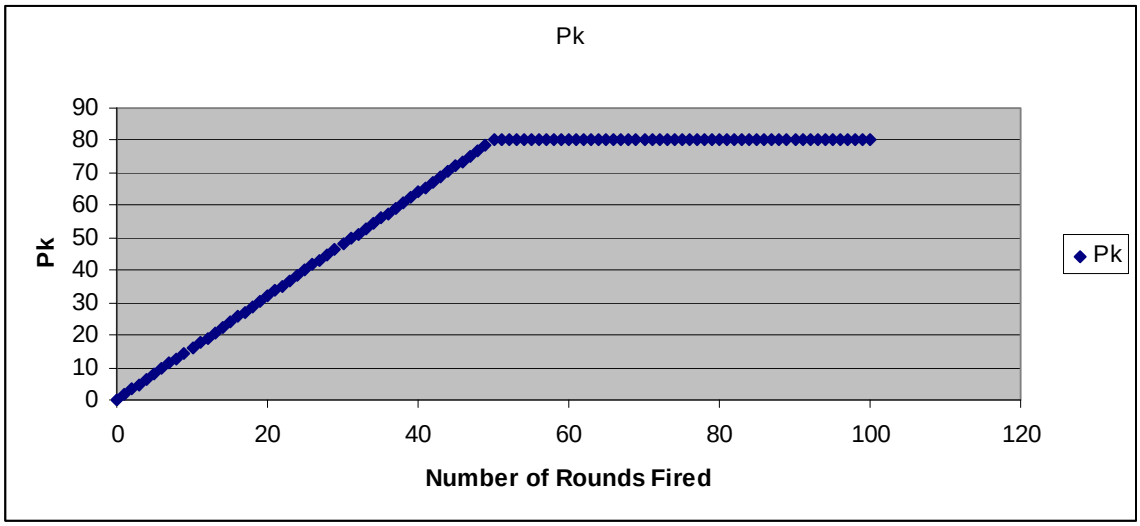


Figure 32 PK Values Computed by Anti-Air Gun Adjudicator

2. Verification of the SAM System, SAM Site and SAM Entities

The logic for the SAM system entity is rather complex, as shown in Figure 18. The approach adopted for verifying the logic on that system is to use different handcrafted scenarios to test the logic progressively. To verify that all the enter ranges, exit ranges, detection, and un-detection are working correctly, a simple scenario consisting of a single SAM site, shown in Figure 33, is crafted. The Pk for the SAM is set to 0.0 intentionally to allow the attacking aircraft to enter and exit the various ranges without being shot down. The printout on the left hand side of Figure 33 displays the system behaving correctly in the scenario.

The next scenario is shown in Figure 34, aims to verify the algorithm of the SAM system in handling multiple targets entering the SAM site, as well as the switching of targets when an earlier lock-on target exits the lock-on range of a SAM site. The printout on the left hand side of the diagram shows the ability of the SAM site to switch its lock from target A301 to A303. Although target A302 was scheduled to be detected, the event was cancelled as soon as it exited the lock-on range. The velocity of A303 has been intentionally slowed down for this scenario to allow time for the SAM site to lock-on to it before it flies out of the lock-on range.

```
Targets:
A301 (240.000,-200.000) [0.000,0.000]
A301 enters lock on range of : Lock-on Range Sensor: 0
A301 entered Bomb Release Line!
A301 enters firing range of: Firing Range Sensor: 0
Lock-on Range Sensor: 0 detected A301!
Lock-on Range Sensor: 0 acquired a lock on A301
Fire Missile
Scaling factor: 0.5491776095959081
Pk: 0.0. Random Number: 27.236883396635125
Target missed
Fire Missile
Scaling factor: 0.1543247268041309
Pk: 0.0. Random Number: 61.89558101034209
Target missed
A301 enters no firing range of No Firing Range Sensor: 0
A301 exits no firing range of No Firing Range Sensor: 0
Fire Missile
Pk: 0.0. Random Number: 91.99837841258537
Scaling factor: 0.032225581003183346
Target missed
Fire Missile
Pk: 0.0. Random Number: 82.29659803827086
Scaling factor: 0.34062439120364446
Target missed
Fire Missile
Pk: 0.0. Random Number: 80.54656879125811
Scaling factor: 0.7477108206682526
Target missed
Fire Missile
Pk: 0.0. Random Number: 37.88240073050443
Scaling factor: 0.40656299504818527
Target missed
A301 exits firing range of: Firing Range Sensor: 0
A301 exits lock on range of: Lock-on Range Sensor: 0
Lock-on Range Sensor: 0 undetects A301!
Entities.SAMSite.17 released lock on target: A301
```

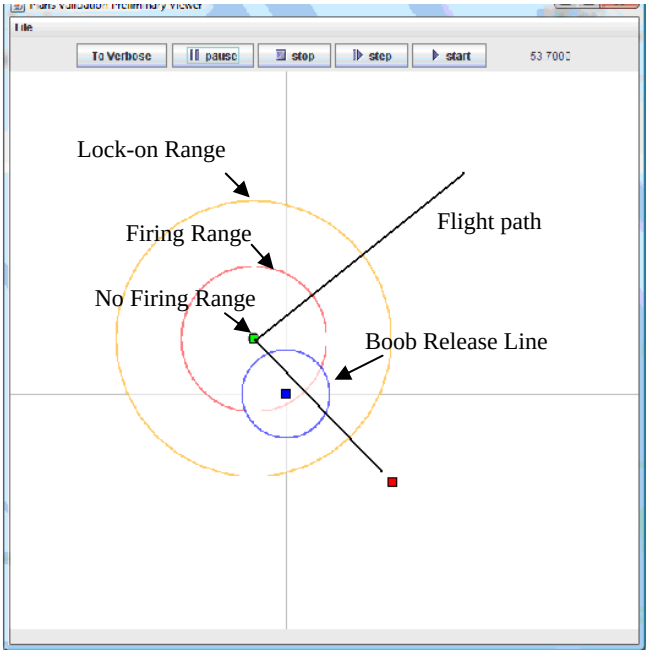


Figure 33 Single SAM Site Scenario for SAM System Verification

```
Targets:
A301 (240.000,-200.000) [0.000,0.000]
A302 (240.000,-180.000) [0.000,0.000]
A303 (240.000,-220.000) [0.000,0.000]
A302 enters lock on range of : Lock-on Range Sensor: 0
A301 enters lock on range of : Lock-on Range Sensor: 0
A303 enters lock on range of : Lock-on Range Sensor: 0
A301 entered Bomb Release Line! 1 leakages so far.
A303 entered Bomb Release Line! 2 leakages so far.
A302 entered Bomb Release Line! 3 leakages so far.
A302 enters firing range of: Firing Range Sensor: 0
A301 enters firing range of: Firing Range Sensor: 0
Lock-on Range Sensor: 0 detected A302!
Lock-on Range Sensor: 0 acquired a lock on A302
A303 enters firing range of: Firing Range Sensor: 0
Lock-on Range Sensor: 0 detected A301!
Sensor is locked-on to another target
Fire Missile
Lock-on Range Sensor: 0 detected A303!
Sensor is locked-on to another target
                    .
                    .
                    .
A301 exits firing range of: Firing Range Sensor: 0
A302 exits lock on range of: Lock-on Range Sensor: 0
Lock-on Range Sensor: 0 undetects A302!
Entities.SAMSite.23 released lock on target: A302
Enter Range event scheduled
A301 exits lock on range of: Lock-on Range Sensor: 0
Lock-on Range Sensor: 0 undetects A301!
Enter Range event scheduled
Lock-on Range Sensor: 0 detected A303!
Lock-on Range Sensor: 0 acquired a lock on A303
Fire Missile
                    .
                    .
                    .
```

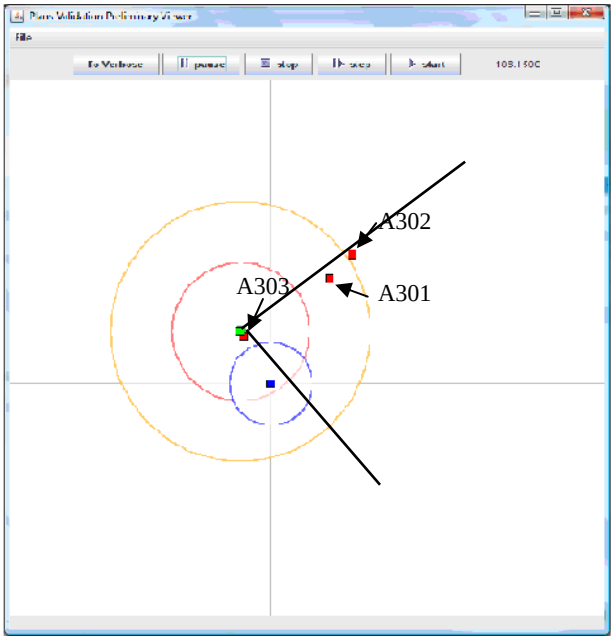


Figure 34 Scenario to Verify Target Switching Algorithm

In typical air defense deployment, SAM sites are often overlapped to prevent gaps in coverage. The following scenario aims to verify the ability for the SAM system to hand over its target from one SAM site to its neighboring SAM site. The scenario setup for this test is shown in Figure 35. From the printout on the left hand side of the figure, it can be seen that as soon as SAMSite 23 releases its lock on A302, SAMSite 30 acquires a lock on it as soon as it is well within the lock-on range of SAMSite 30. The same sequence is repeated for A303. SAMSite 30 did not acquire a lock on A301 because in the model, a SAM site is only able to acquire a lock on one aircraft at a time, and since A302 and A301 are flying in very close proximity, and a lock on has already been acquired on A302, the SAM site is unable to acquire another lock. This is the correct behavior that the system is designed for.

```
Targets:
A301 (240.000,-200.000) [0.000,0.000]
A302 (240.000,-180.000) [0.000,0.000]
A303 (240.000,-220.000) [0.000,0.000]
A302 enters lock on range of.; Lock-on Range Sensor: 0
A301 enters lock on range of.; Lock-on Range Sensor: 0
A303 enters lock on range of.; Lock-on Range Sensor: 0
A301 entered Bomb Release Line! 1 leakages so far.
A303 entered Bomb Release Line! 2 leakages so far.
A302 entered Bomb Release Line! 3 leakages so far.
                .
                .
                .
A302 exits firing range of: Firing Range Sensor: 0
A301 enters lock on range of.; Lock-on Range Sensor: 1
A301 exits firing range of: Firing Range Sensor: 0
Lock-on Range Sensor: 1 detected A302!
Target has been locked by another sensor.
A302 enters firing range of: Firing Range Sensor: 1
A302 exits lock on range of: Lock-on Range Sensor: 0
Lock-on Range Sensor: 0 undetects A302!
Entities.SAMSite.23 released lock on target: A302
Enter Range event scheduled for A301 in SamSite Entities.SAMSite.23
Lock-on Range Sensor: 1 acquired a lock on A302
Lock-on Range Sensor: 1 detected A301!
Sensor is locked-on to another target
A301 enters firing range of: Firing Range Sensor: 1
A301 exits lock on range of: Lock-on Range Sensor: 0
Lock-on Range Sensor: 0 undetects A301!
Enter Range event scheduled for A303 in SamSite Entities.SAMSite.23
A303 enters lock on range of.; Lock-on Range Sensor: 1
Lock-on Range Sensor: 0 detected A303!
Lock-on Range Sensor: 0 acquired a lock on A303
A303 exits firing range of: Firing Range Sensor: 0
Lock-on Range Sensor: 1 detected A303!
Target has been locked by another sensor.
A302 exits firing range of: Firing Range Sensor: 1
A301 exits firing range of: Firing Range Sensor: 1
A302 exits lock on range of: Lock-on Range Sensor: 1
Lock-on Range Sensor: 1 undetects A302!
Entities.SAMSite.30 released lock on target: A302
Enter Range event scheduled for A301in SamSite Entities.SAMSite.30
A303 enters firing range of: Firing Range Sensor: 1
A301 exits lock on range of: Lock-on Range Sensor: 1
Lock-on Range Sensor: 1 undetects A301!
Enter Range event scheduled for A303in SamSite Entities.SAMSite.30
A303 exits lock on range of: Lock-on Range Sensor: 0
Lock-on Range Sensor: 0 undetects A303!
Entities.SAMSite.23 released lock on target: A303
Lock-on Range Sensor: 1 detected A303!
Lock-on Range Sensor: 1 acquired a lock on A303
A303 enters no firing range of No Firing Range Sensor: 1
A303 exits no firing range of No Firing Range Sensor: 1
```

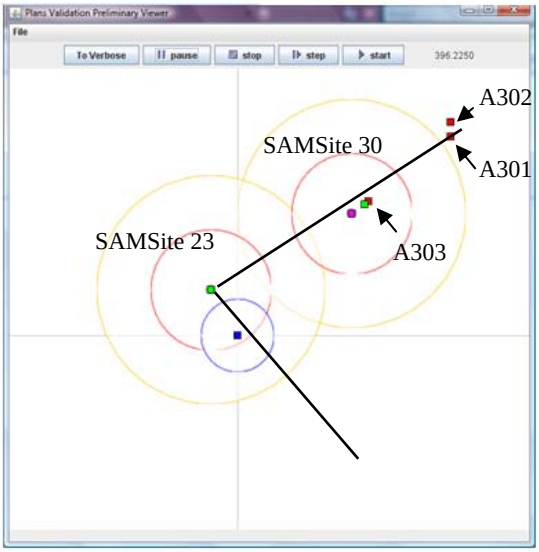


Figure 35 Scenario for Verification of Target Handover Between SAM Sites

The last scenario for SAM system verification is to verify that a SAM site is able to switch its lock to another target flying within the lock-on range as soon as an earlier locked aircraft is shot down. The scenario for this verification is similar to the previous case, except that the Pk value of the SAM is set to non-zero value so that targets will get destroyed in order to trigger the handover process. The scenario is as shown in Figure 36. In the scenario, all targets were killed and it can be seen from the printout that the sensor is able to switch its lock to the next target once the locked target is destroyed. This is the correct behavior.

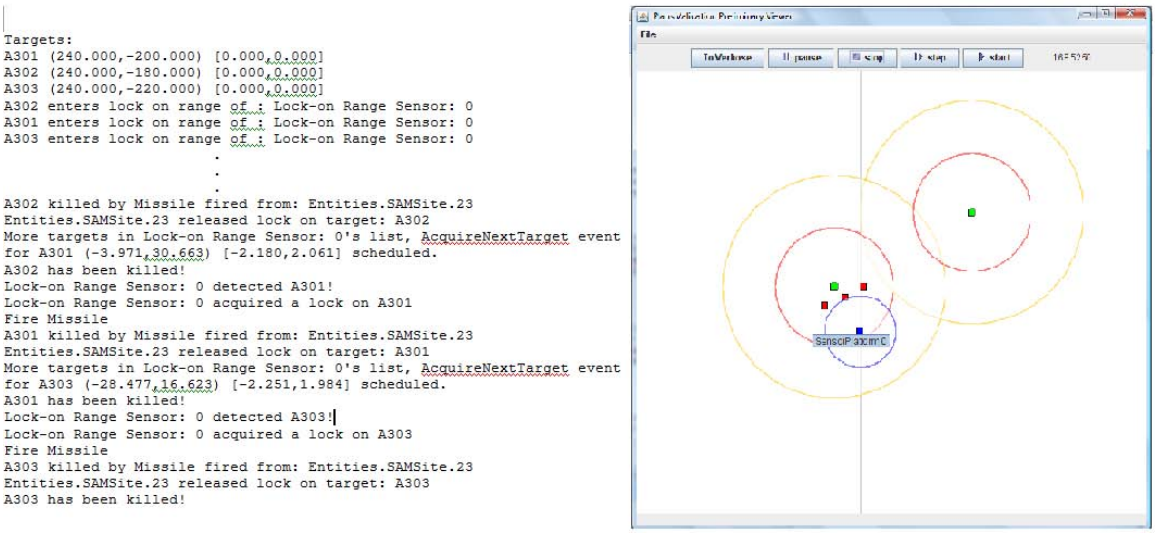


Figure 36 Scenario for Verification of Target Handover Within SAM Site

3. Verification of Gun Site and Gun Entities

The verification approach for the gun site and gun entities is similar to the approach taken for the SAM system. As in the SAM system case, scenarios were hand crafted for testing the logic of gun site and gun entities. The first scenario shown in Figure 37 aims to verify the logic for enter/exit firing range events, gun engage events, adjudicate events and the reload events shown in Figure 24. In this scenario, the gun's Pk was set to 0.0 for the target to trigger the enter/exit firing range and the reload events.

From the printout, it can be seen that the reload event is triggered after 100 rounds, which is the configured magazine size for the AAG. The gun and gun site are behaving in the correct way.

```
Targets:
A301 (240.000,-200.000) [0.000,0.000]
A301 entered Bomb Release Line! 1 leakages so far.
GunSite: Gun Detection!
Rounds fired: 52
Gun #0 engaging target A301
Gun Pk: 0.0 | Random number: 36.52685751794513
Target missed
Rounds fired: 48
Gun #0 engaging target A301
Gun Pk: 0.0 | Random number: 40.9065356409703
Gun #0 empty. Gun reloaded.
Target missed
Rounds fired: 59
Gun #0 engaging target A301
Gun Pk: 0.0 | Random number: 58.76405545157778
Target missed
Rounds fired: 41
Gun #0 engaging target A301
Gun Pk: 0.0 | Random number: 28.684455591056192
Gun #0 empty. Gun reloaded.
Target missed
A301 exits firing range of: Gun #0
Gun #0 disengaged A301
```

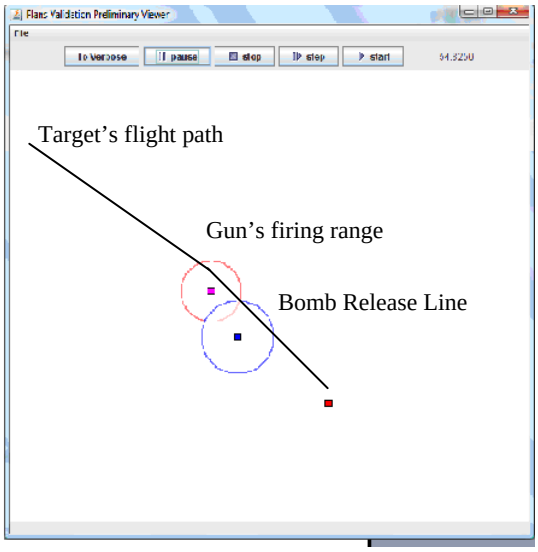


Figure 37 Scenario for Verification Anti-Air Gun

The second scenario shown in Figure 38 aims to verify the ability of the gun to switch to other targets in the firing range once the current target flies out of range. From the printout, it can be seen that the gun disengaged A302 and was assigned a new target A301 when A302 flies out of the gun's firing range. The gun is behaving correctly.

```
Targets:
A301 (240.000,-200.000) [0.000,0.000]
A302 (220.000,-180.000) [0.000,0.000]
A302 entered Bomb Release Line! 1 leakages so far.
A301 entered Bomb Release Line! 2 leakages so far.
GunSite: Gun Detection!
Rounds fired: 63
Gun #0 engaging target A302
Gun Pk: 0.0 | Random number: 2.5840898268143175
Target missed
Rounds fired: 37
Gun #0 engaging target A302
Gun Pk: 0.0 | Random number: 55.035294722985405
Gun #0 empty. Gun reloaded.
Target missed
GunSite: Gun Detection!
Gun #0 is busy.
Rounds fired: 92
Gun #0 engaging target A302
Gun Pk: 0.0 | Random number: 34.07686313896101
Target missed
Rounds fired: 8
Gun #0 engaging target A302
Gun Pk: 0.0 | Random number: 69.55627646918485
Gun #0 empty. Gun reloaded.
Target missed
A302 exits firing range of: Gun #0
New target assigned to Gun #0
Gun #0 disengaged A302
Rounds fired: 58
Gun #0 engaging target A301
Gun Pk: 0.0 | Random number: 58.879306752424924
Target missed
Rounds fired: 42
Gun #0 engaging target A301
Gun Pk: 0.0 | Random number: 1.4554608759263288
Gun #0 empty. Gun reloaded.
Target missed
A301 exits firing range of: Gun #0
Gun #0 disengaged A301
```

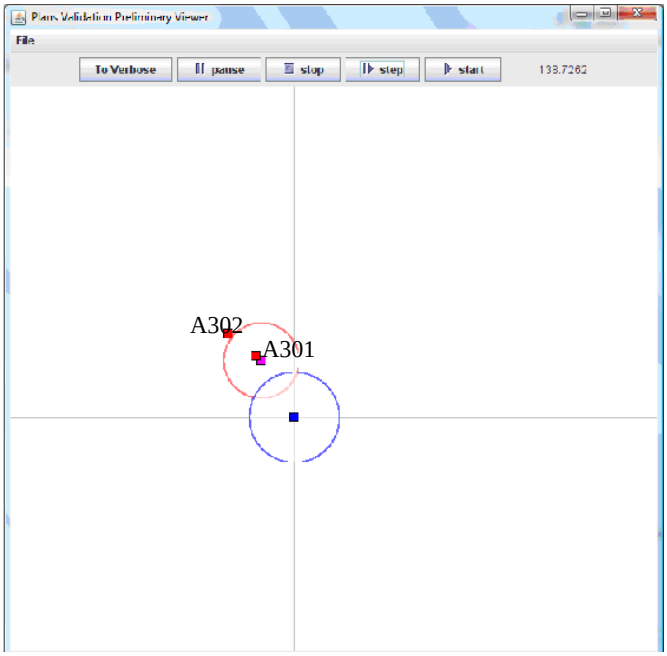


Figure 38 Scenario for Target Switching When a Target Flies Out-of-Range

The last scenario, as shown in Figure 39, aims to verify that the AAG is able to switch to other targets in the firing range when the current target is destroyed. The Pk was set to its normal range in this scenario in order for targets to be destroyed. From the printout, it can be seen that once A302 is killed, the gun disengages A302 and starts to engage A301. The gun is behaving correctly.

```
Targets:
A301 (230.000,-190.000) [0.000,0.000]
A302 (220.000,-180.000) [0.000,0.000]
A302 entered Bomb Release Line! 1 leakages so far.
A301 entered Bomb Release Line! 2 leakages so far.
GunSite: Gun Detection!
GunSite: Gun Detection!
Rounds fired: 98
Gun #0 engaging target A302
Gun Pk: 80.0 | Random number: 6.339501503295364
A302 has been killed!
Gun #0 disengaged.
Rounds fired: 2
Gun #0 engaging target A301
Gun Pk: 0.0 | Random number: 40.0416091716361
Gun #0 empty. Gun reloaded.
Target missed
Rounds fired: 92
Gun #0 engaging target A301
Gun Pk: 80.0 | Random number: 37.209526573556204
A301 has been killed!
Gun #0 disengaged.
```

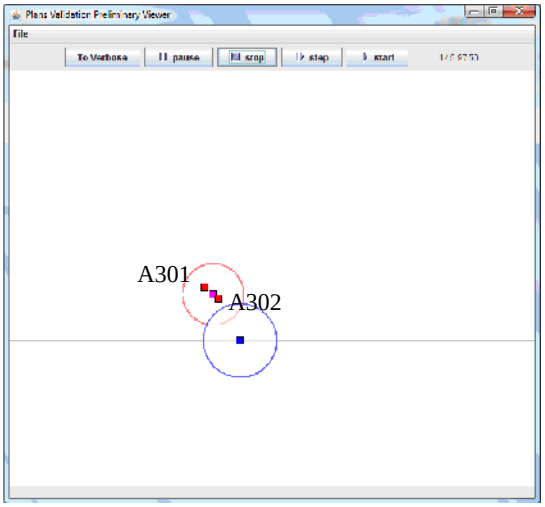


Figure 39 Scenario for Target Switching When a Target is Killed

B. VERIFICATION OF AGENT MODEL

In order to conduct verification on the agent model described in the previous chapter, there is a need to create an air defense scenario and test it against the DES engine.

The agent modeling application and its algorithm are developed using Java based on multi-threading design. Multi-threading design is used to facilitate simultaneous agents processing and communication. The method for message exchange between the agent modeling application and the DES is via Java Object Serialization using User Datagram Protocol (UDP).

The agent application supports an area of operation of 50km by 50km for the air-defense scenario, which forms an area of 2500km^2. The figure below shows the logical system diagram of agent model and discrete event simulator.

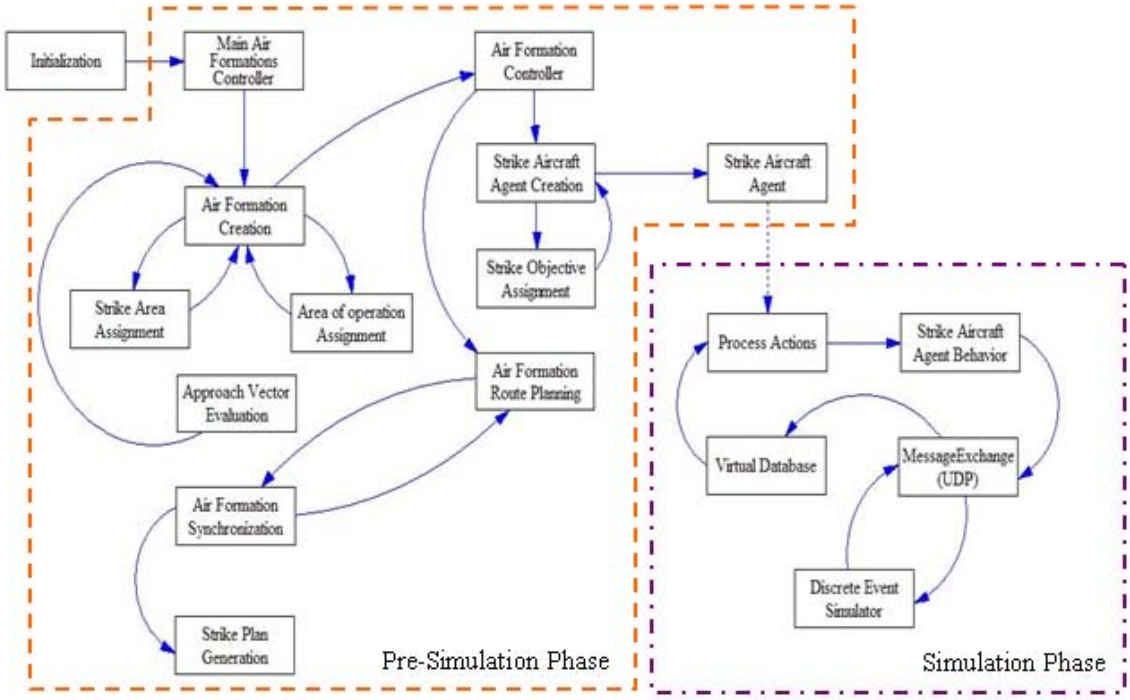


Figure 40 Logical System Diagram of Agent Model and Discrete Event Simulator

The orange dotted box shows the computation process for generation of the strike plan by the Main Air Formation Controller; this is done during the pre-simulation phase. After the strike plan is generated, this plan is then read by the discrete event simulator to start the simulation of the strike aircrafts towards the air defense area.

The air defense plan is shown in Figure 41. At the initialization phase, the parameters shown in Table 2 were fed into the agent model application. With these parameters, the main air formation controller was able to create the appropriate number of participating air formation controller, which in turn, the air formation controller will create the number of agent strike aircraft within the formation.

At the main air formation controller, the position evaluation technique was used calculate the best approach vectors. The number of approach vectors to calculate depends upon how many air formations were created.

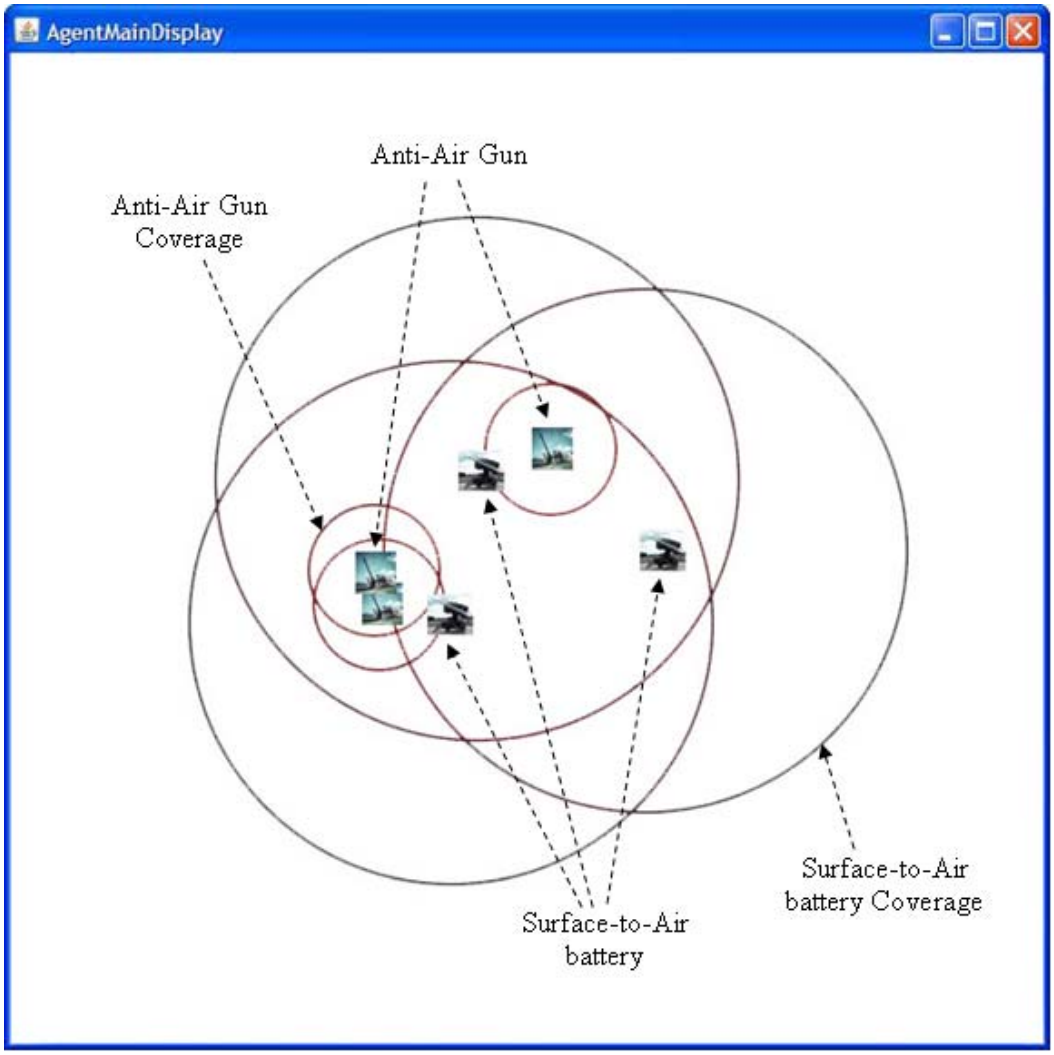


Figure 41 Air Defense Plan with SAM Site and Gun Deployment

Initialization Parameters
Air defense Plan
Air Defense Artillery Profile
Area of operation
Number of Strike Aircraft
Type of Strike Aircraft
Initial Setup Position

Table 2 System Initialization Parameters

The scoring for a best approach vector using the position evaluation technique is based on three factors:

- Speed of the aircraft (Default set at Mach 1) (km/hr)
- Distance exposed to air defense projected from the strike objective (km)
- Exposure duration to the air defense unit from the strike objective (Seconds)

The figure below shows how the best approach vector is selected based on the 3 factors previously described.

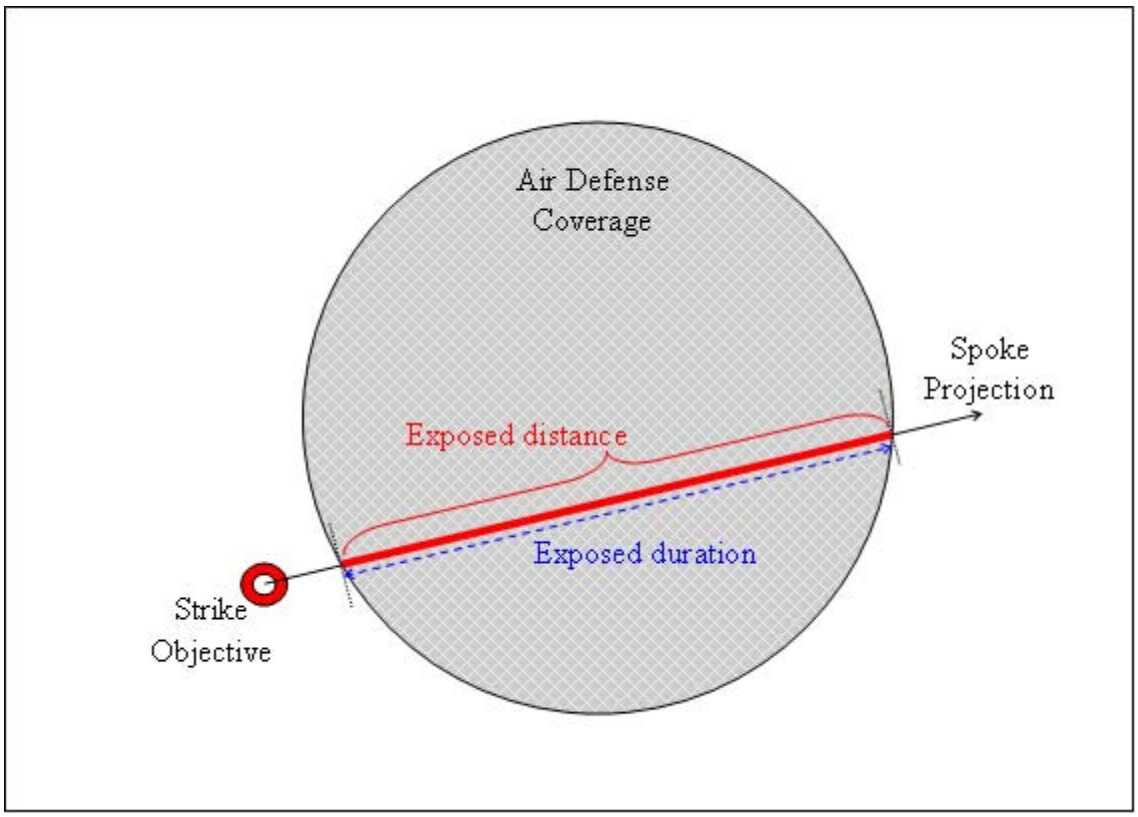


Figure 42 Score Calculation for Best Approach Vector

The actual projection of spoke, to calculate the score to find suitable approach vector to the air defenses, is shown in Figure 43. The main air-formations controller agent

is tasked to find two approaches to the air defense plan, and has chosen the two most suitable approach vectors, which are highlighted in blue, for the air formations to use to approach the strike objective.

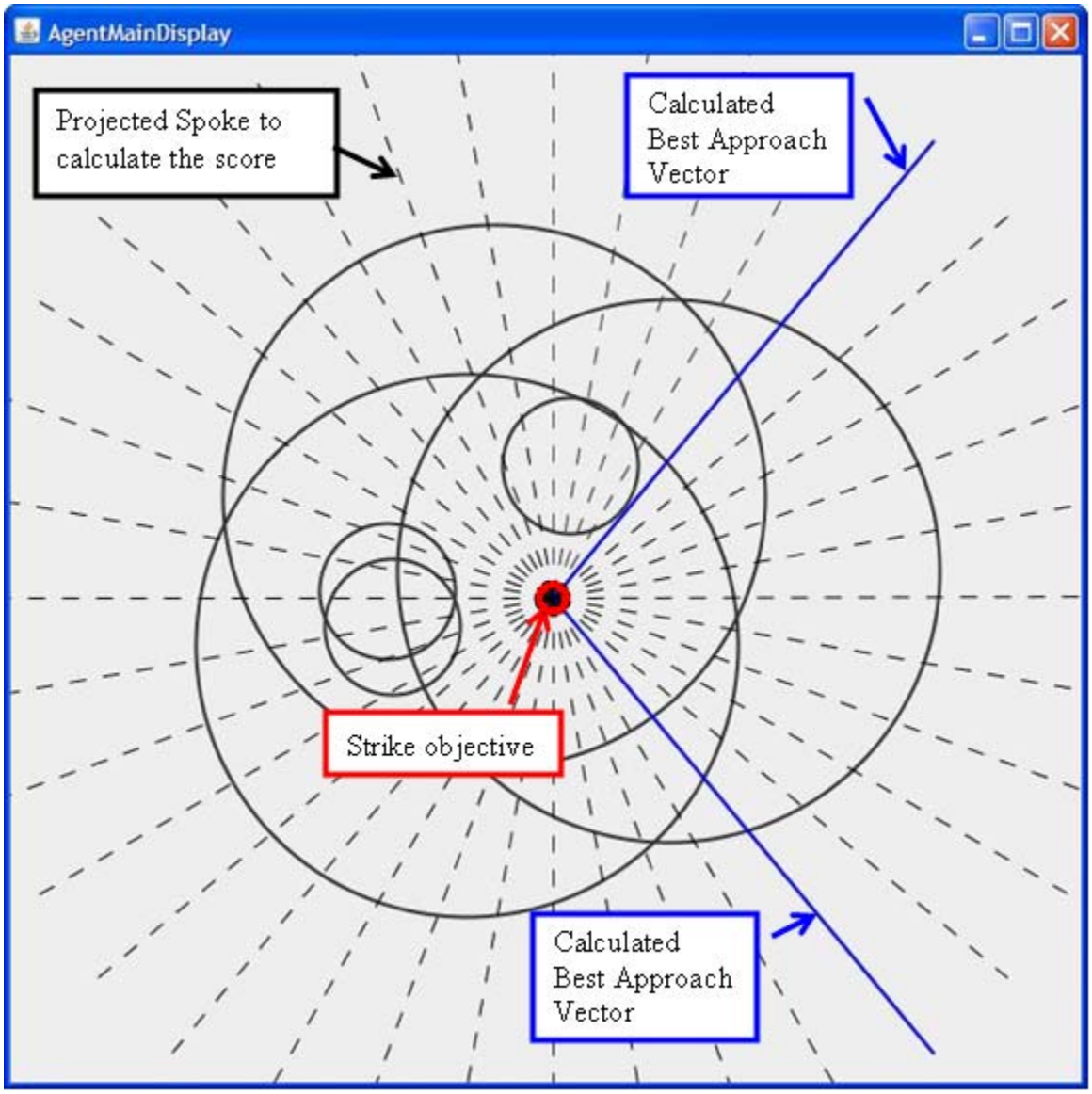


Figure 43 Projected Spokes for Calculation of Best Approach Vector

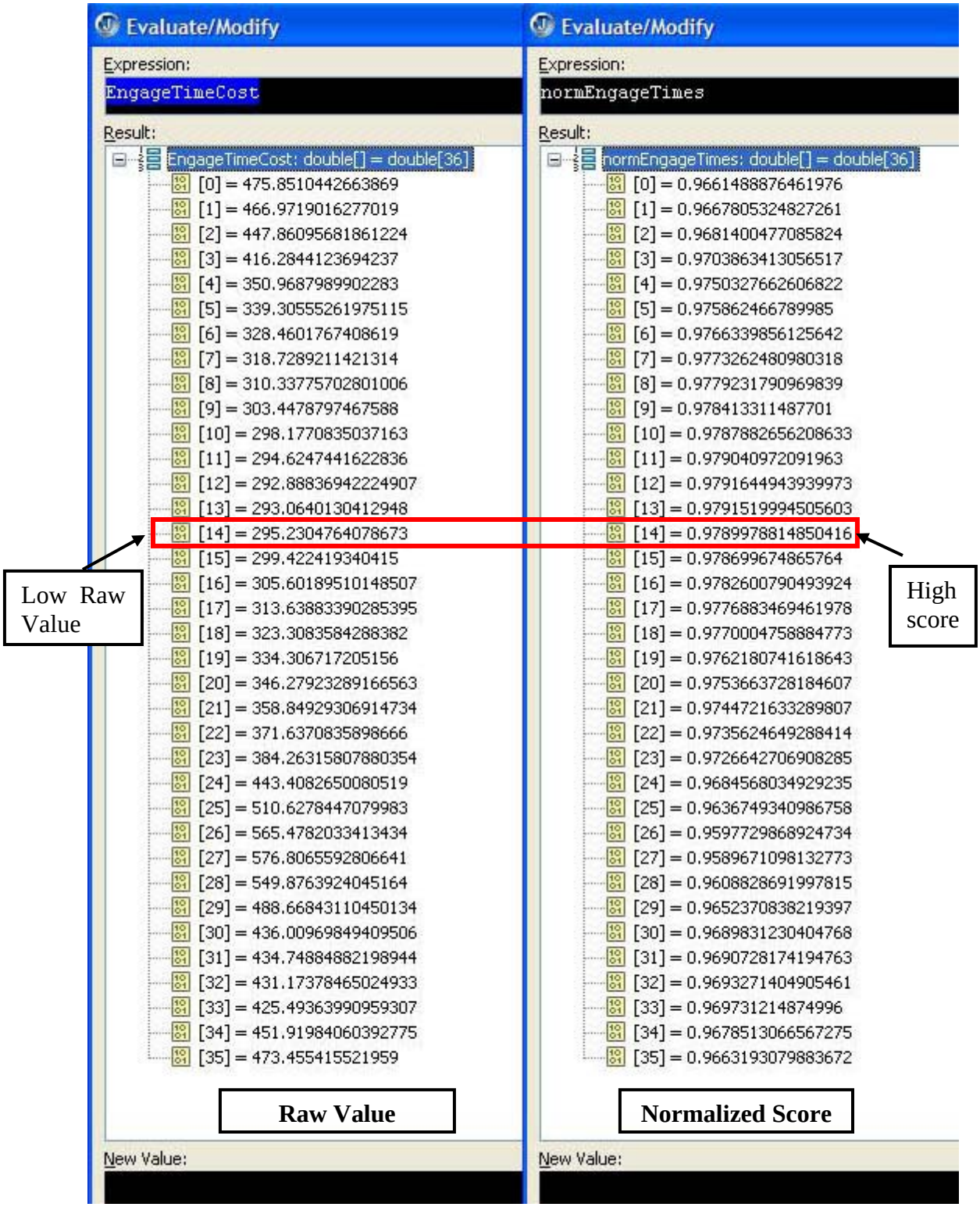


Figure 44 Exposed Duration Score for Each Projected Spoke of the Air Defense Plan

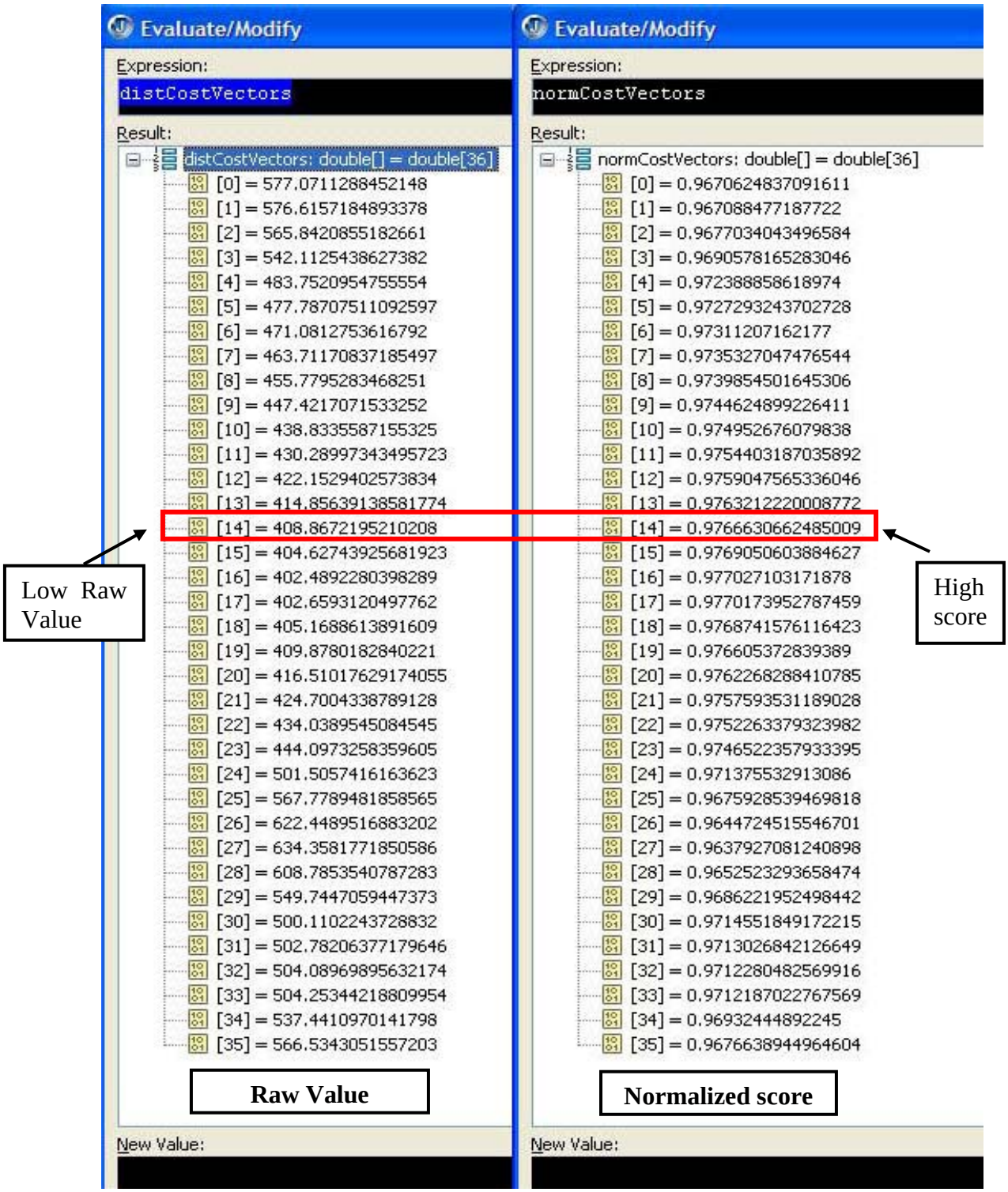


Figure 45 Exposed Distance Score for Each Projected Spoke of the Air Defense Plan

The next step after finding suitable approach vectors is to pass the information to each individual air formation agent to plan the best route to the strike objective. The path finding algorithm for the air formation agent uses cell-based decomposition methods for real world abstraction. Each cell, forming the abstraction of the real world, is represented as a pixel which is equivalent to 200 meters distance in real world. The entire map abstraction for the path finding has 40000 cells representing the equivalent of 40km by 40km in the real world.

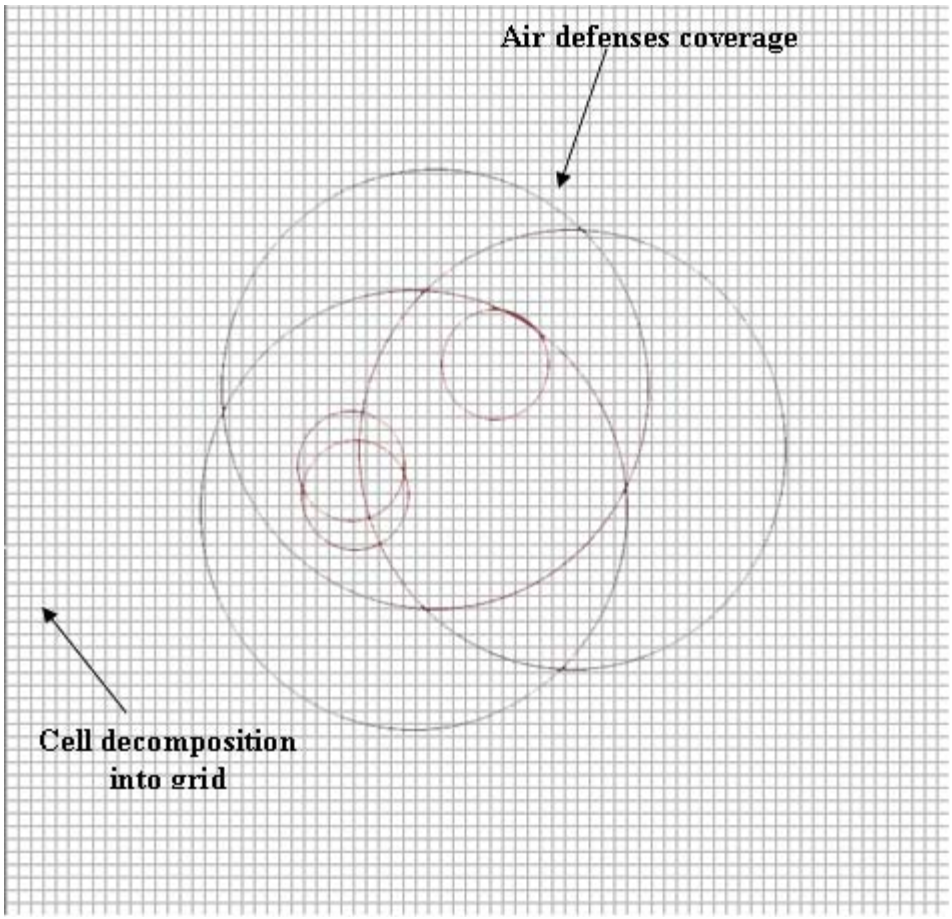


Figure 46 Zoomed-In View of the Cell Decomposition Search Space

In the path-planning algorithm, different cost structures are assigned to different types of air defense units. Units that are deemed as high risk, such as Surface-to-Air Missile (SAM) site, are assigned a higher cost value to the area covered by the SAM site,

as compared to units such as Anti-Air Gun site. The cost of overlapping areas by air defense units is the summation cost of individual air defense units and there is also a need to assign a minimal cost of movement for neutral area as well. The current assignment of the cost for different type of SAM site and the minimal cost of movement are chosen by the trial and error method as there is no one correct way of determining the “correct” values to be use, it is found that if the cost difference between two type of SAM site is too high, this will cause the system to spend a longer time searching for the optimal path.

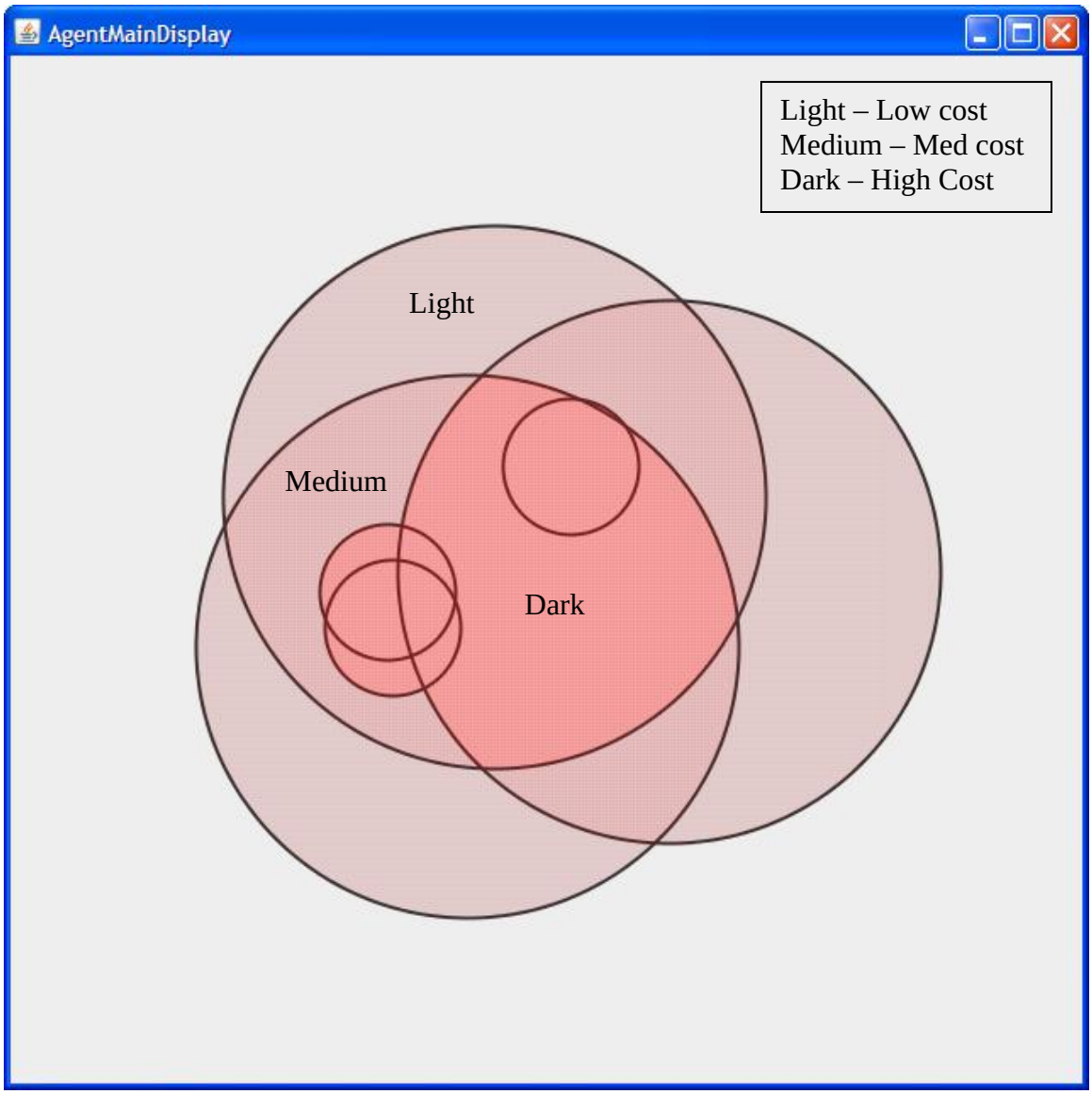


Figure 47 Air Defense Coverage Represent as Color Tone

Once the various initial costs of the air defense areas are calculated and assigned to the cell which is shown in the figure, above indicating various levels of the costs by the depth of the color, the algorithm will proceed with planning a route to the strike objective, taking into account the approach vectors assigned to the air-formation and cost of penetrating the air defenses coverage.

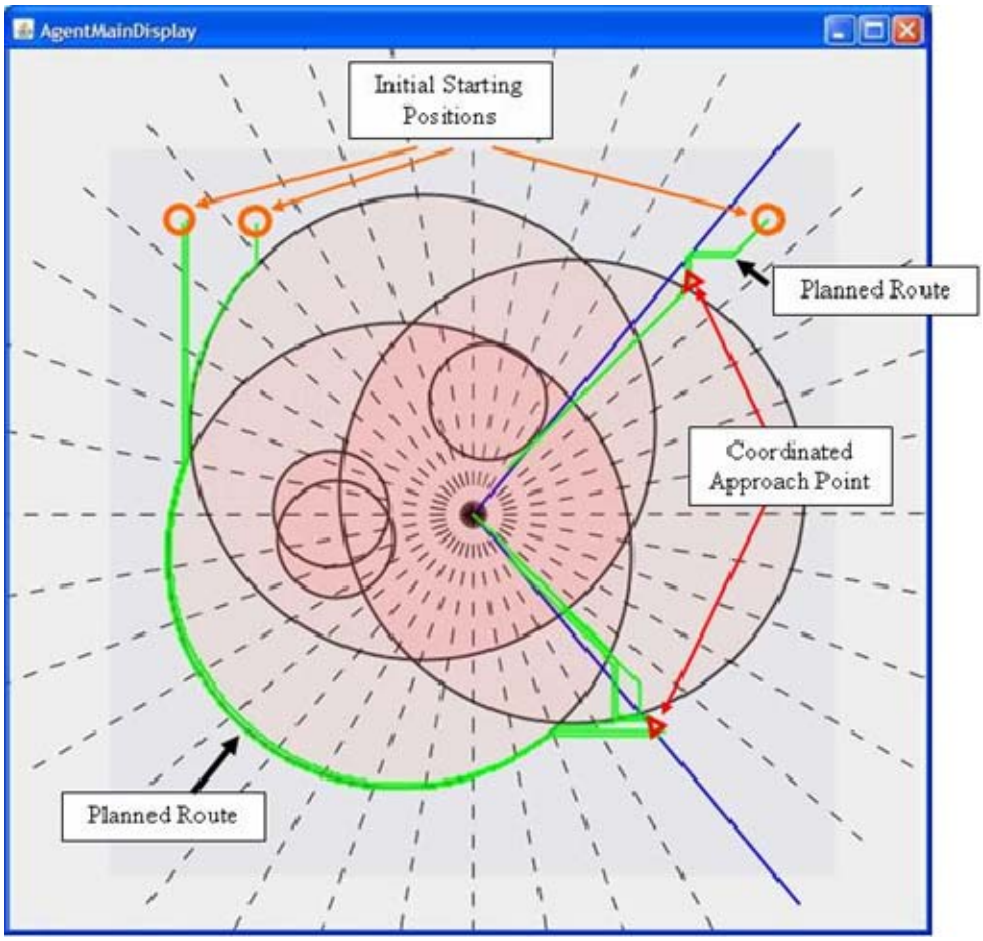


Figure 48 Completed Strike Plan Generation for 3 Air Formations

The generated strike plan will be parsed into the simulation engine for execution of the attack plan. During the simulation run, updated messages from the simulator will be sent to the agent application using Java Object Serialization via UDP. The simulator will send the following information to the strike aircraft agents in the agent application for processing:

Inputs Messages to Strike Aircraft Agent	
POSITIONAL	Positional updates of the agent in the simulation environment
RADAR LOCK ON	Alert agent aircraft of Radar lock on
RADAR LOCK OFF	Alert agent aircraft of Radar lost lock
INCOMING SA MISSILE	Alert agent aircraft of incoming surface-to-air missile
ANTI-AIR GUN FIRING	Alert agent aircraft of Anti-Air gun firing

Table 3 Input Messages to Strike Aircraft Agent

The strike aircraft agent will then use the information from the simulator to generate an action, if any, depending on which message type is received back to the simulator.

Agent Action Message
Update Agent status (Aircraft status & Environment changes)
EVASIVE ACTION
STRIKE ACTION

Table 4 Agent Action Message

The Evasive action is executed by the agent when it received an alert message of type “Incoming SA Missile” from the simulator. The agent will then proceed to check it current heading and determine a another heading away from the current heading, once a new heading is obtain, the agent will then plot a path towards the head by waypoints. The current implementation is for the agent to plot a straight line distance away from the current heading. The agent will send the new path to the simulator for in order for the simulation to plot the agent path away from the incoming missile.

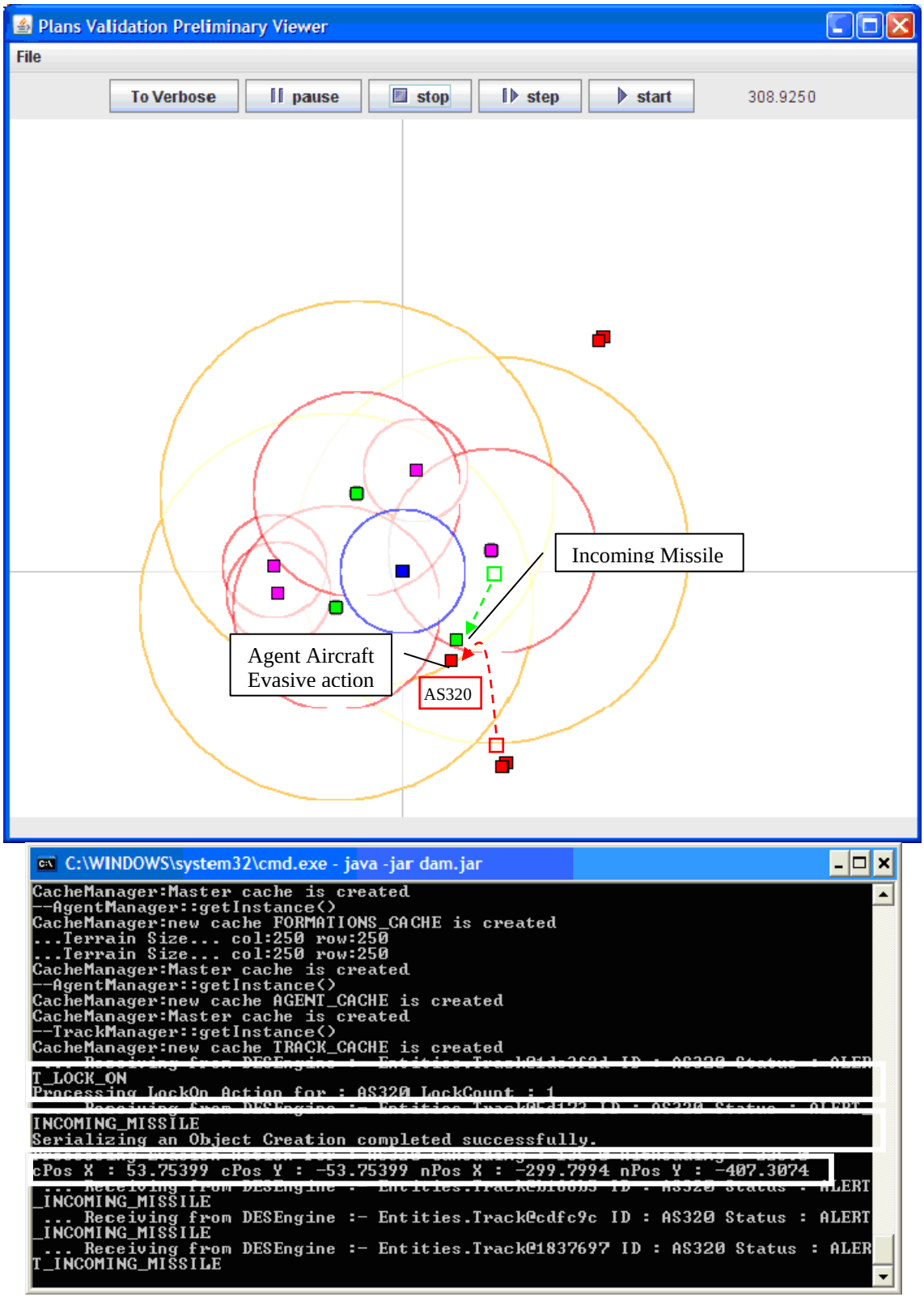


Figure 49 Evasive Action by Agent Aircraft AS320

The Strike Action is executed by the agent after it has completed its Evasive action or received an alert message of type “Radar Lock Off”. The agent will then plot a path towards the strike area and will send the path waypoints to the simulator so that the simulator can plot the agent path towards the strike area.

C. SYSTEM VALIDATION

The validation methodology adopted for the DES agent application and the Discrete Event Simulator are face validation via functional decomposition with the subject-matter-expert (SME). The functional decomposition approach is to validate the system at component levels, rather than at a system level, and within this functional decomposition, face validation approach is done by seeking SME views to validate the system components.

1. DES Agent Application

In the agent application, the approach vector selection via positional evaluation and the path planning algorithms are validated. The approach vector selection is desired to achieve the following:

Approach Vector Selection
GAP EXPLOITATION
LEAST AIR_DEFENSE COVERAGE EXPLOITATION

Table 5 Goals of Approach Vector Selection

The approach vectors generation for the agent application is able to generate the approaches shown in Figure 3 and Figure 4 based on the two vector approach selection criteria as illustrated in Figure 50.

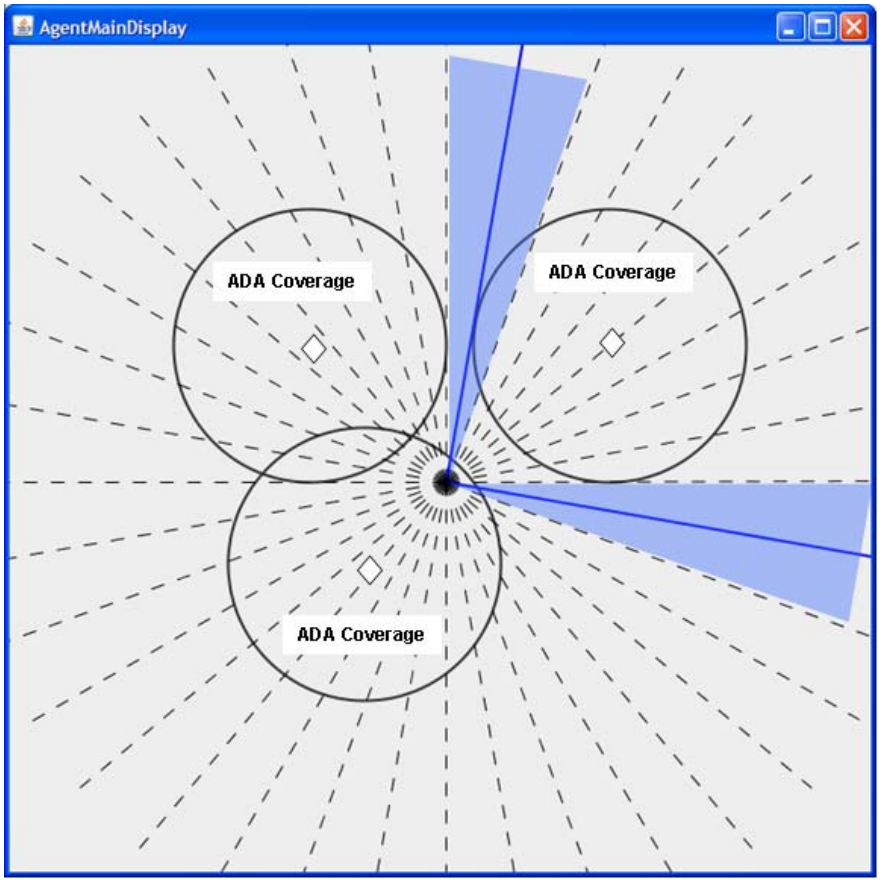


Figure 50 Agent Generated Gap Exploitation

Likewise, the least air-defense coverage exploitation is illustrated in Figure 51.

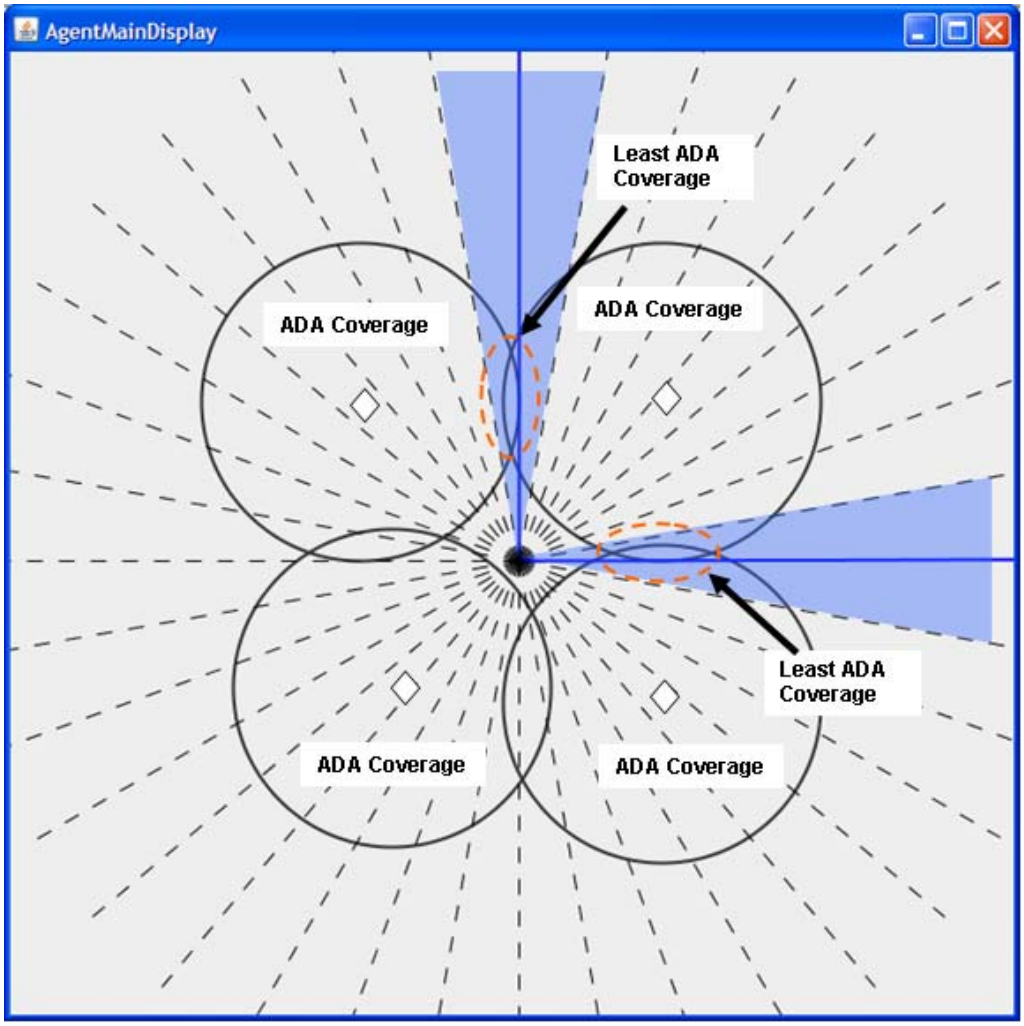


Figure 51 Agent Generated Least Air-Defense Coverage Exploitation

The DES Agent application is able to determine the least air-defense coverage and hence able to exploit it as attack vectors for the strike formation. Both of these results were shown to the SME, a fighter squadron commander, and the SME is satisfied with the results and agreed that the objectives for the approach vectors generation were met.

For the path planning generated by the DES agent application, taking into account the air defense coverage only, is shown below:

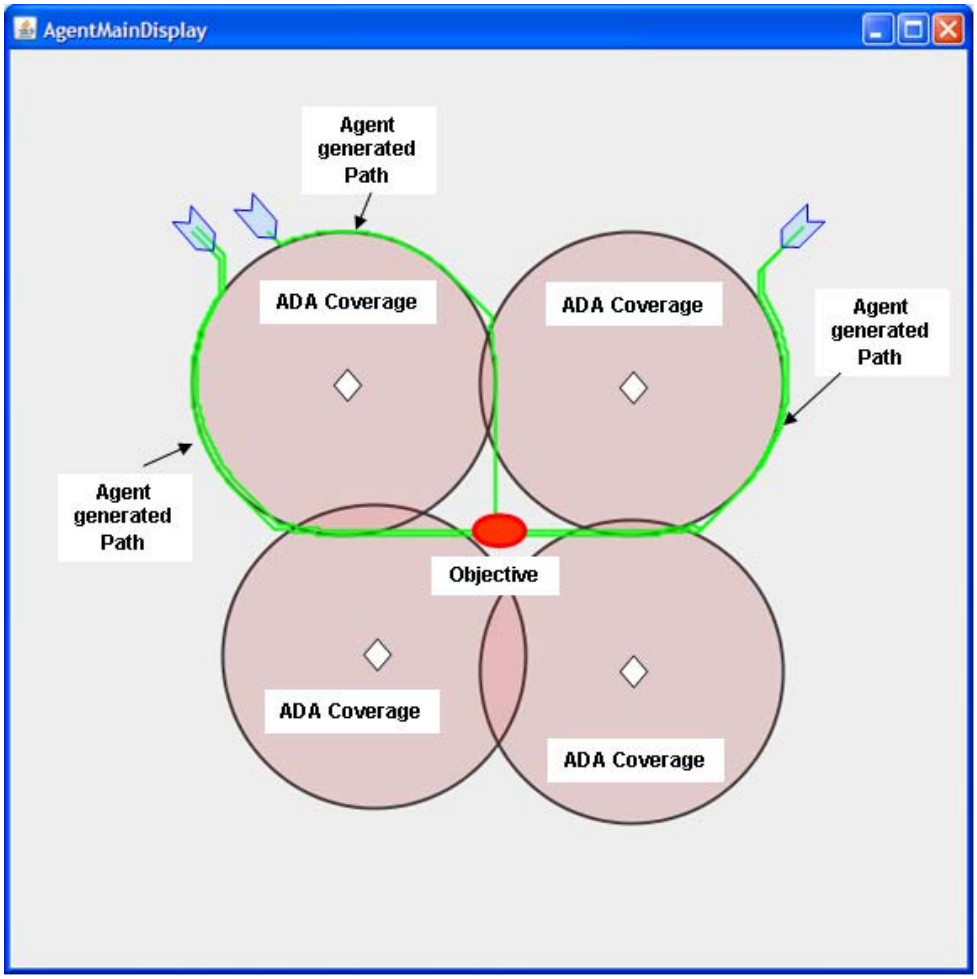


Figure 52 Path Planning Generated by DES Agent Application

The path planned by the DES agent is able to circumvent the air defense zone, however, the actual flight path generate by SME is more of a waypoints path where each leg of the waypoint is a straight line. A more refined path smoothing algorithm is required to “straighten” the path.

Using an actual air defense plan produced by a SME, and with both of these components integrated together to generate the strike plan, this takes into account the air defense area and approach vectors. The agent application is able to generate the following strike plan:

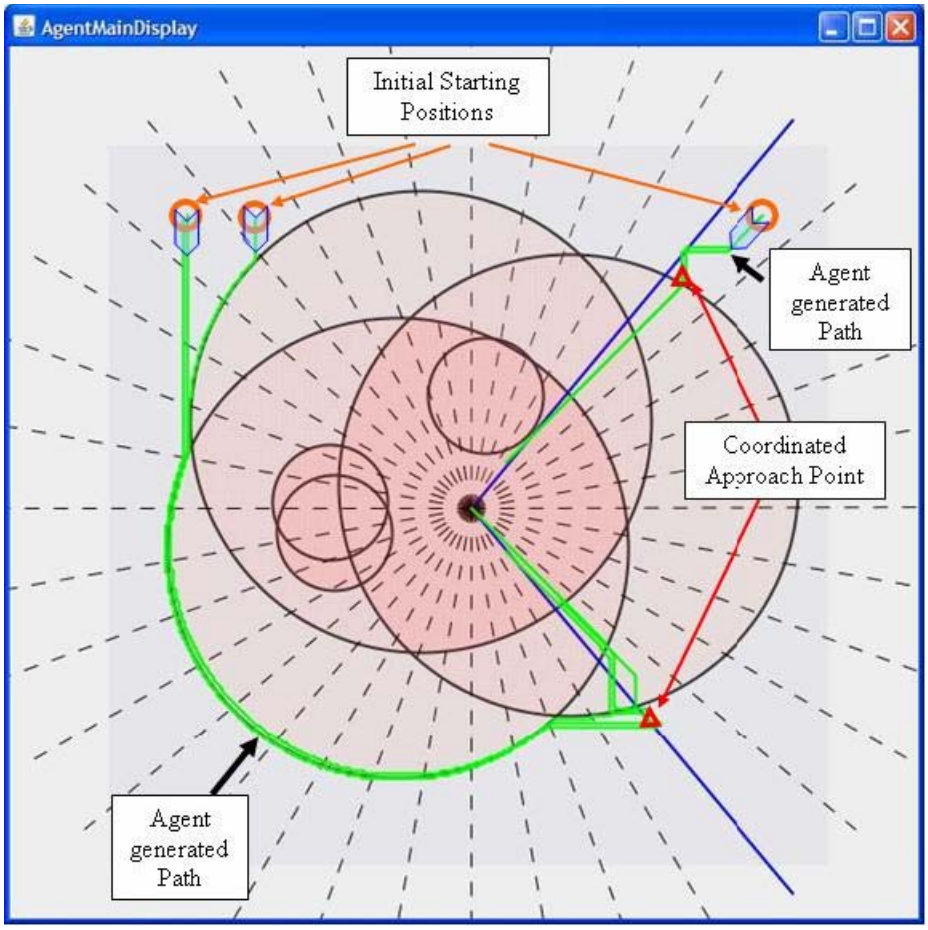


Figure 53 Strike Plan Generated by DES Agent Application

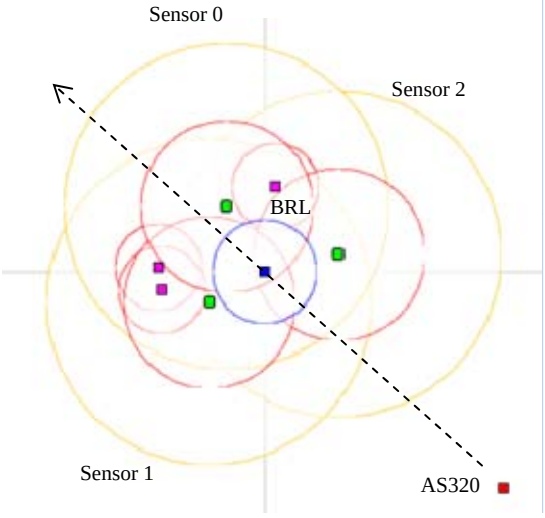
For the result shown in Figure 53, the SME reiterate that if the planned path is for strike mission a more "straighten" path using waypoints as legs of the path to the objective is more correct, however, the SME highlighted that if the current path generated is for SEAD mission or Decoy mission, it is conceivable to have this type of path.

2. Validation of DES Engine

The DES engine validation process adopts the functional decomposition approach, which starts with simple scenarios and builds up to the eventual scenario shown in Figure 58. The simple scenarios are geared towards validating the behavior of the air defense system's response to individual and small number of aircrafts.

a. Scenario 1: Single Aircraft Flying Across SAM Systems

The scenario and the console output file is shown in Figure 54. In this scenario, a single aircraft is flown across the BRL as shown. The Pk for all the weapon systems has been set to zero, so that the aircraft is able to fly across and test the response of all the systems without being killed. The SME agrees with the response of the system based on the doctrine to engage the first aircraft that enters the lock-on range.

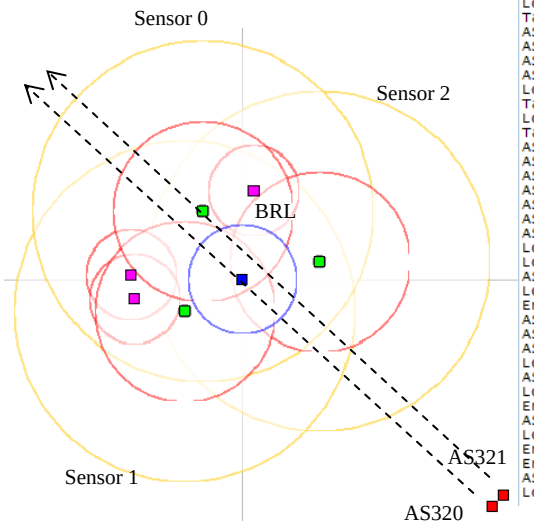



```
AS320 enters lock on range of : Lock-on Range Sensor: 2
AS320 enters lock on range of : Lock-on Range Sensor: 1
Lock-on Range Sensor: 2 detected AS320!
Lock-on Range Sensor: 2 acquired a lock on AS320
AS320 enters firing range of: Firing Range Sensor: 2
Fire Missile
AS320 enters lock on range of : Lock-on Range Sensor: 0
Missile fired from: Entities.SAMSite.43 (47.718,-42.296) [0.000,0.000]
AS320 (47.620,-43.291) [-1.850,1.682]
Pk: 0.0. Random Number: 84.39768758864659
Target missed
Fire Missile
AS320 entered Bomb Release Line! 1 leakages so far.
Lock-on Range Sensor: 1 detected AS320!
Target has been locked by another sensor.
Missile fired from: Entities.SAMSite.43 (26.198,-22.492) [0.000,0.000]
AS320 (25.686,-23.350) [-1.850,1.682]
Pk: 0.0. Random Number: 20.489461575733937
Target missed
Fire Missile
AS320 enters firing range of: Firing Range Sensor: 0
Lock-on Range Sensor: 0 detected AS320!
Target has been locked by another sensor.
Missile fired from: Entities.SAMSite.43 (2.020,-0.547) [0.000,0.000]
AS320 (1.156,-1.051) [-1.850,1.682]
Pk: 0.0. Random Number: 14.533177939940122
Target missed
Fire Missile
AS320 exits firing range of: Firing Range Sensor: 2
Missile fired from: Entities.SAMSite.43 (-20.445,21.559) [0.000,0.000]
AS320 (-21.438,21.438) [-1.414,1.414]
Pk: 0.0. Random Number: 69.81195431008163
Target missed
AS320 exits firing range of: Firing Range Sensor: 1
AS320 exits lock on range of: Lock-on Range Sensor: 2
Lock-on Range Sensor: 2 undetects AS320!
Entities.SAMSite.43 released lock on target: AS320
Lock-on Range Sensor: 1 acquired a lock on AS320
AS320 exits firing range of: Firing Range Sensor: 0
AS320 exits lock on range of: Lock-on Range Sensor: 1
Lock-on Range Sensor: 1 undetects AS320!
Entities.SAMSite.36 released lock on target: AS320
Lock-on Range Sensor: 0 acquired a lock on AS320
AS320 exits lock on range of: Lock-on Range Sensor: 0
Lock-on Range Sensor: 0 undetects AS320!
Entities.SAMSite.29 released lock on target: AS320
```

Figure 54 Scenario 1 and its Output File

b. Scenario 2: Double Aircraft Flying Across SAM Systems

In the scenario shown in Figure 55, an additional aircraft is added on top of scenario one, to validate the behavior of the SAM system's response to more than one aircraft entering their lock-on range. The missile firing events have been removed intentionally to keep the output file concise, to illustrate the sequence of lock acquisition of the various sensors clearly. The SME agrees with the response of the system but commented that, depending on doctrines that vary from country to country, some SAM sites might not engage receding aircraft to conserve resources for next wave of aircraft.




```
AS321 enters lock on range of : Lock-on Range Sensor: 2
AS320 enters lock on range of : Lock-on Range Sensor: 2
AS320 enters lock on range of : Lock-on Range Sensor: 1
AS321 enters lock on range of : Lock-on Range Sensor: 1
Lock-on Range Sensor: 2 detected AS321!
Lock-on Range Sensor: 2 acquired a lock on AS321
AS321 enters firing range of: Firing Range Sensor: 2
Lock-on Range Sensor: 2 detected AS320!
Sensor is locked-on to another target
AS320 enters firing range of: Firing Range Sensor: 2
AS321 enters lock on range of : Lock-on Range Sensor: 0
AS320 enters lock on range of : Lock-on Range Sensor: 0
AS320 entered Bomb Release Line! 1 leakages so far.
Lock-on Range Sensor: 1 detected AS320!
Lock-on Range Sensor: 1 acquired a lock on AS320
AS321 entered Bomb Release Line! 2 leakages so far.
Lock-on Range Sensor: 1 detected AS321!
Target has been locked by another sensor.
AS320 enters firing range of: Firing Range Sensor: 1
AS321 enters firing range of: Firing Range Sensor: 1
AS321 enters firing range of: Firing Range Sensor: 0
AS320 enters firing range of: Firing Range Sensor: 0
Lock-on Range Sensor: 0 detected AS321!
Target has been locked by another sensor.
Lock-on Range Sensor: 0 detected AS320!
Target has been locked by another sensor.
AS320 exits firing range of: Firing Range Sensor: 2
AS321 exits firing range of: Firing Range Sensor: 2
AS321 exits firing range of: Firing Range Sensor: 1
AS321 enters no firing range of No Firing Range Sensor: 0
AS321 exits no firing range of No Firing Range Sensor: 0
AS320 exits firing range of: Firing Range Sensor: 1
AS320 exits lock on range of: Lock-on Range Sensor: 2
Lock-on Range Sensor: 2 undetects AS320!
Lock-on Range Sensor: 0 acquired a lock on AS320
AS321 exits lock on range of: Lock-on Range Sensor: 2
Lock-on Range Sensor: 2 undetects AS321!
Entities.SAMSite.46 released lock on target: AS321
AS320 exits firing range of: Firing Range Sensor: 0
AS321 exits firing range of: Firing Range Sensor: 0
AS321 exits lock on range of: Lock-on Range Sensor: 1
Lock-on Range Sensor: 1 undetects AS321!
AS320 exits lock on range of: Lock-on Range Sensor: 1
Lock-on Range Sensor: 1 undetects AS320!
Entities.SAMSite.39 released lock on target: AS320
AS320 exits lock on range of: Lock-on Range Sensor: 0
Lock-on Range Sensor: 0 undetects AS320!
Entities.SAMSite.32 released lock on target: AS320
Enter Range event scheduled for AS321 in SamSite Entities.SAMSite.32
AS321 exits lock on range of: Lock-on Range Sensor: 0
Lock-on Range Sensor: 0 undetects AS321!
```

Figure 55 Scenario 2 and its Output File

c. Scenario 3: Double Aircraft Flying Across AA Gun Sites

In this scenario shown in Figure 56, two aircraft were made to fly across two AA gun sites to validate the behavior of the gun sites' response to multiple aircraft flying into their firing range. Similar to the earlier scenario, the Pk for the AA guns has been set to zero to allow the aircraft to fly across the gun sites. The SME agrees with the response of the gun sites based on the assumption that the AA guns are not guided by any radar, and thus, it would make sense to direct more fire power on a single target. He added that for radar guided gun sites, a common practice is to assign a gun to each aircraft, due to its higher precision. This could also prevent AA guns from being taken out by other aircraft in the formation that are not being fired at.

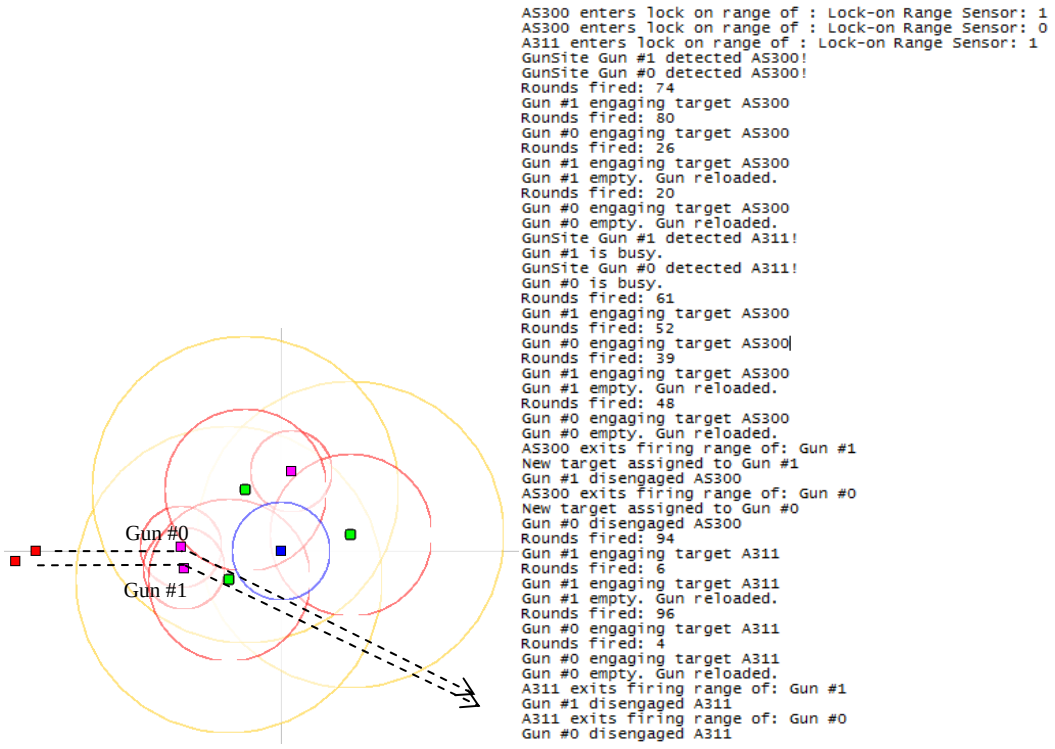


Figure 56 Scenario 3 and its Output File

d. Scenario 4: Aircraft Flying Across SAM and AA Gun Sites

In this scenario shown in Figure 57, the Pk for the weapon systems were configured to their vendor specified values. The three groups of aircraft were made to fly across the defended area in three approach vectors. This scenario tests the overall response of the air defense system with a “dummy” plan, whereby the aircraft do not try to exploit weaknesses in the defense system and fly in “blind” into the area. The interesting point about this scenario is that AS320 is made to pick up its velocity after it enters the lock-on range. As a result, it overtakes AS310 and enters the firing range (red circle) earlier than AS310. The output is not shown due to the amount of events that have occurred. The SME agrees with the behavior of the overall system. He noticed the behavior of AS320 in the simulation run and commented that the approach taken by aircraft AS310 and AS320 is a “clever” way to defeat a SAM system which engages its target based on the first-come-first-serve basis. This could be an area that the SAM system could be made to behave smarter.

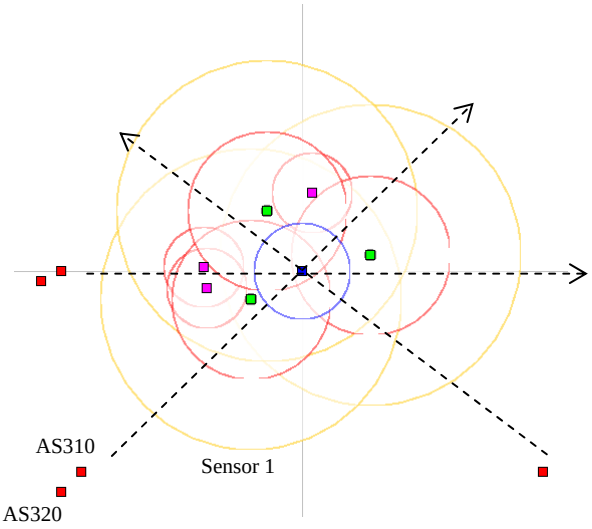


Figure 57 Scenario 4 and its Output File

e. Scenario 5: Realistic Scenario

In this final scenario, the system is validated using a realistic air defense plan, crafted by the SME, and the strike plan generated by the DES Agent Application as shown in Figure 58. The strike plan generated by the DES Agent is also being assessed by the SME to be a reasonably “smart” plan that is able to exploit the weaknesses in the air defense. In addition, the aircraft in this scenario also has the ability to perform evasive maneuvers based on the messages they received from the DES Engine.

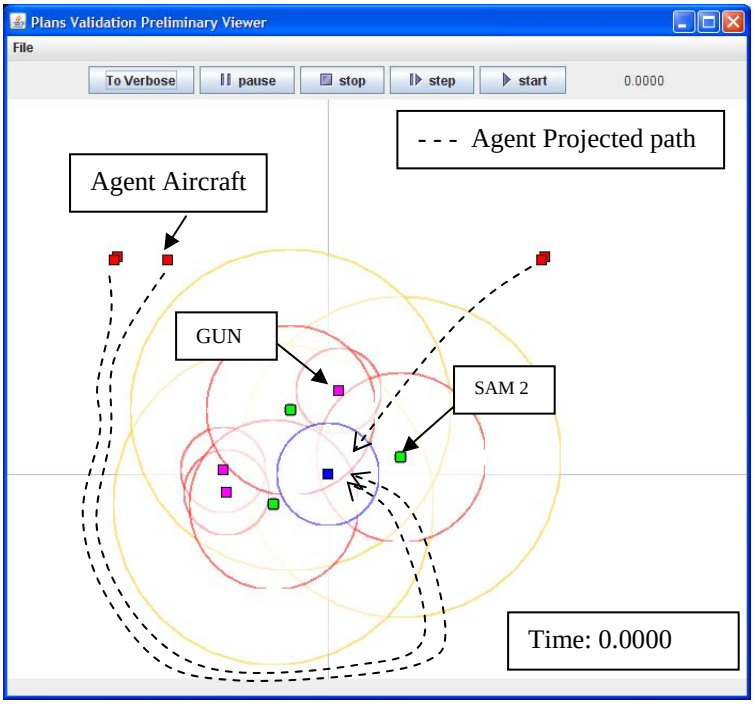


Figure 58 Air Defense Plan and Strike Plan Read and Parsed by DES Engine

As the decoy aircraft agent penetrates into the air-defense zone of the SAM site, the agent will begin to receive “LOCK ON” message and as the agent aircraft proceeds further into the “Firing” zone of the SAM site, it will receive “INCOMING MISSILE” message from the simulator. Figure 59 illustrates this.

The SME agrees with the response of the system on the whole but commented that in a typical SHORAD scenario, most of the aircraft would have been taken out by medium to long range air defenses and it is unlikely to have so many aircrafts flying into the defended area as what is being simulated in the scenario. Thus, in this worst-case situation, it is unavoidable to have a higher number of leakages given the fact that SAM2 is overwhelmed by the aircraft.

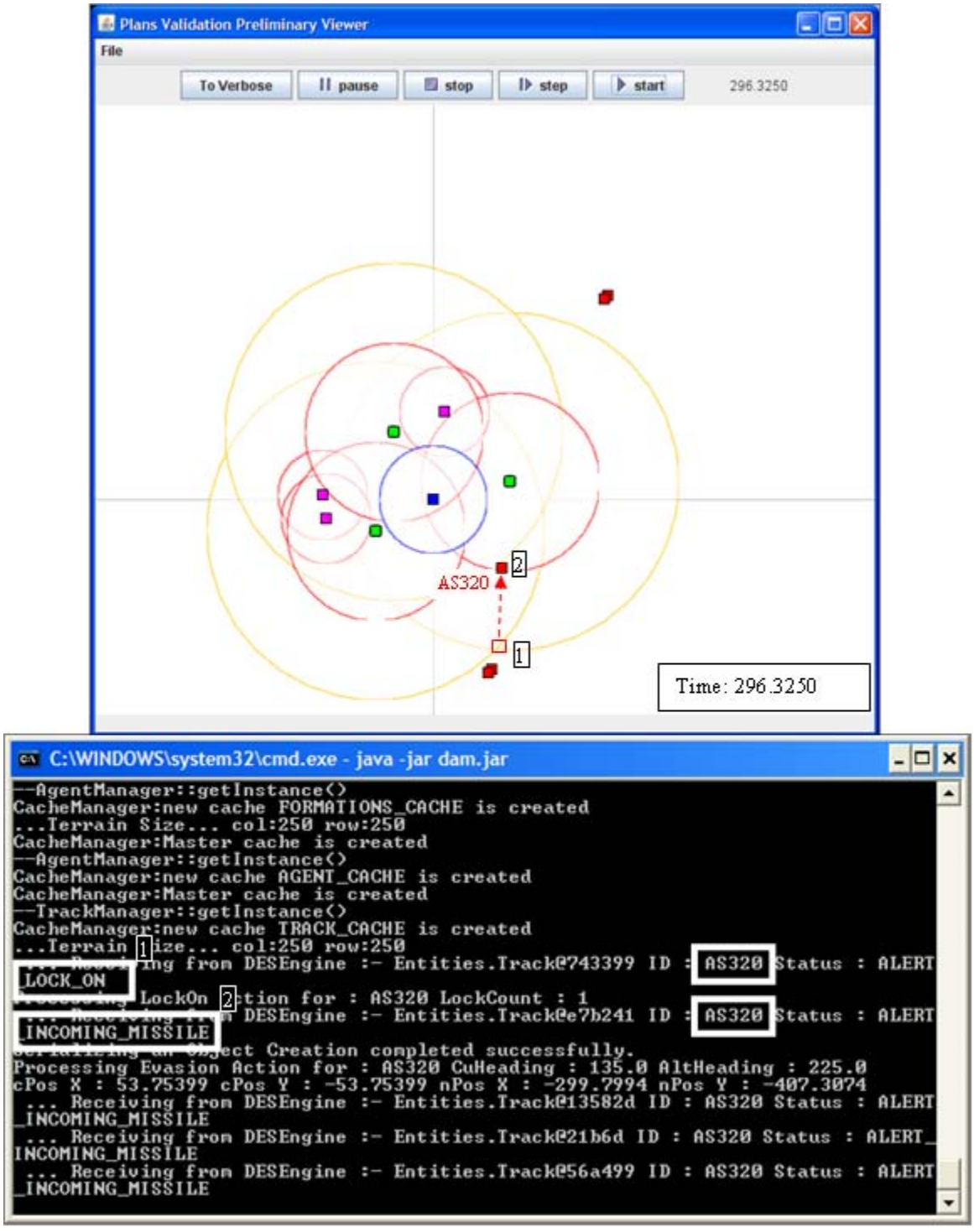


Figure 59 DES Engine Simulating Aircraft Entering Lock-on and Firing Range

IV. EXPERIMENTATION, DATA COLLECTION AND ANALYSIS

In this chapter, the system that was verified and validated in the previous chapter is used to answer questions posed in two scenarios: scenario one is posed by the defender that is defending a key installation; and scenario two is posed by the attacker who is conducting air strikes on the key installation.

In the defender's scenario shown in Figure 60, it is assumed that the defender has no intelligence on the attacker's plan and the route of advance is chosen arbitrarily to test the effectiveness of its weapon systems. In this scenario, the locations of the deployed air defense assets are constrained by the terrain and logistics. The defender wishes to find the weapons properties (which he can later purchase) that will minimize the number of leakage. The question posted is: "What kind of weapon systems should be used in the defense layout to achieve the lowest leakage?"

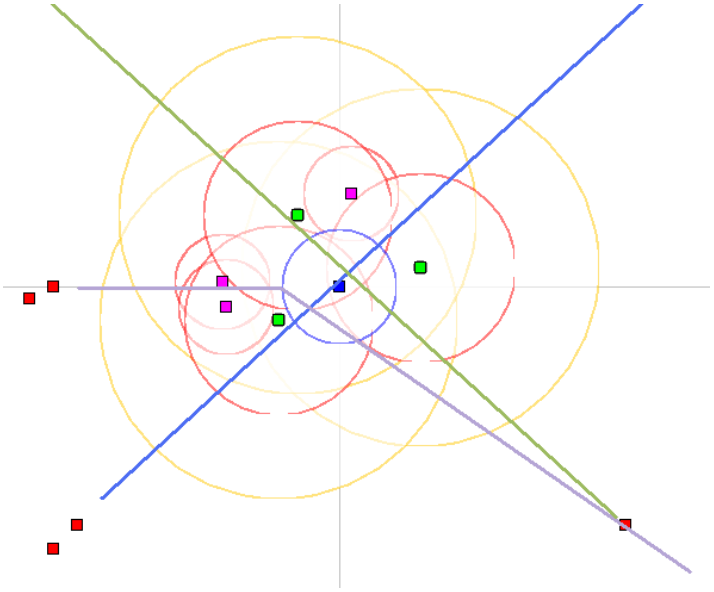


Figure 60 The Defender's Scenario

In the attacker's scenario shown in Figure 61, it is assumed that the attacker has good intelligence on the locations and types of weapon systems deployed, and has planned an attack route using the agent-based tool that would select the safest route of

approach. The question posted is: “How sensitive is our current attack plan to the variation in the weapon systems?” Such variations are caused by imperfect intelligence which is common in any intelligence gathering process. The MOE is based on the number of leakages.

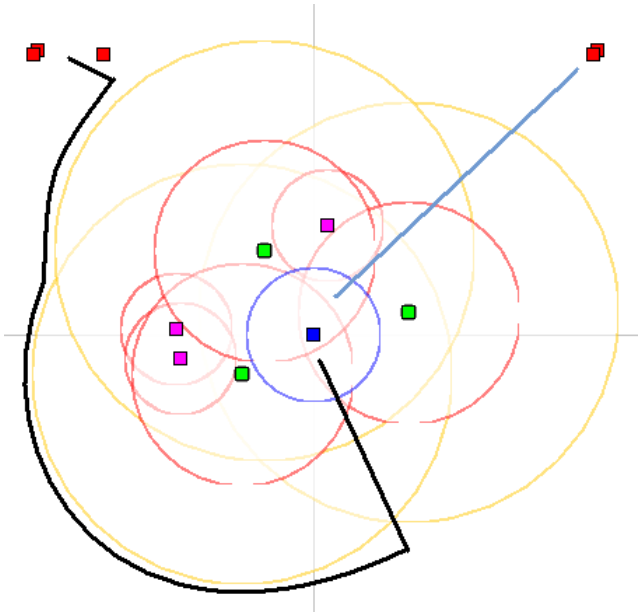


Figure 61 The Attacker's Scenario

1. Design of Experiment

Based on the current model, there are a total of 15 potential main effects that could affect the effectiveness of the air defense plan. The 15 potential main effects and the range over which each effect could be varied are shown in Table 6.

To conduct an experiment with 15 potential effects which has continuous values for most of the effects will be very time consuming. For instance, using a conservative estimation of full factorial 2 level design, there will be 2^15 = 32768 design points. If 50 runs are conducted for each design point, a total of 32768 x 50 = 1,638,400 runs will be required.

This thesis has adopted the Nearly Orthogonal Latin Hypercube (NOLH) by Cioppa & Lucas [14] for the experimental design. Using the NOLH spreadsheet [15], a total of 65 design points were identified, which has drastically reduced the number of runs required from 1,638,400 runs to 3250 runs (65 x 50). The 65 design points are shown in Table 9 in Appendix B. The scatterplot matrix for the design points are shown in Figure 62.

Potential Effects	Min	Max	Units
SAM Reaction Time	5.0	60.0	s
SAM Loading Time	120.0	360.0	s
Missile Per Launcher	2	8	
SAM Min Range	5.0	10.0	'00 m
SAM Max Range	50.0	150.0	'00 m
SAM Max Speed	5.5	8.0	'00 m/s
SAM Engagement Time	3.0	10.0	s
SAM SSKP	55	90	%
Gun Reaction Time	4.0	8.0	s
Gun Loading Time	30.0	60.0	s
Gun Min Range	1.0	5.0	'00 m
Gun Max Range	30.0	40.0	'00 m
Gun Rate of Fire	600	1200	
Gun Magazine Size	200	400	
Gun SSKP	75	85	%

Table 6 Potential Main Effects of the Model

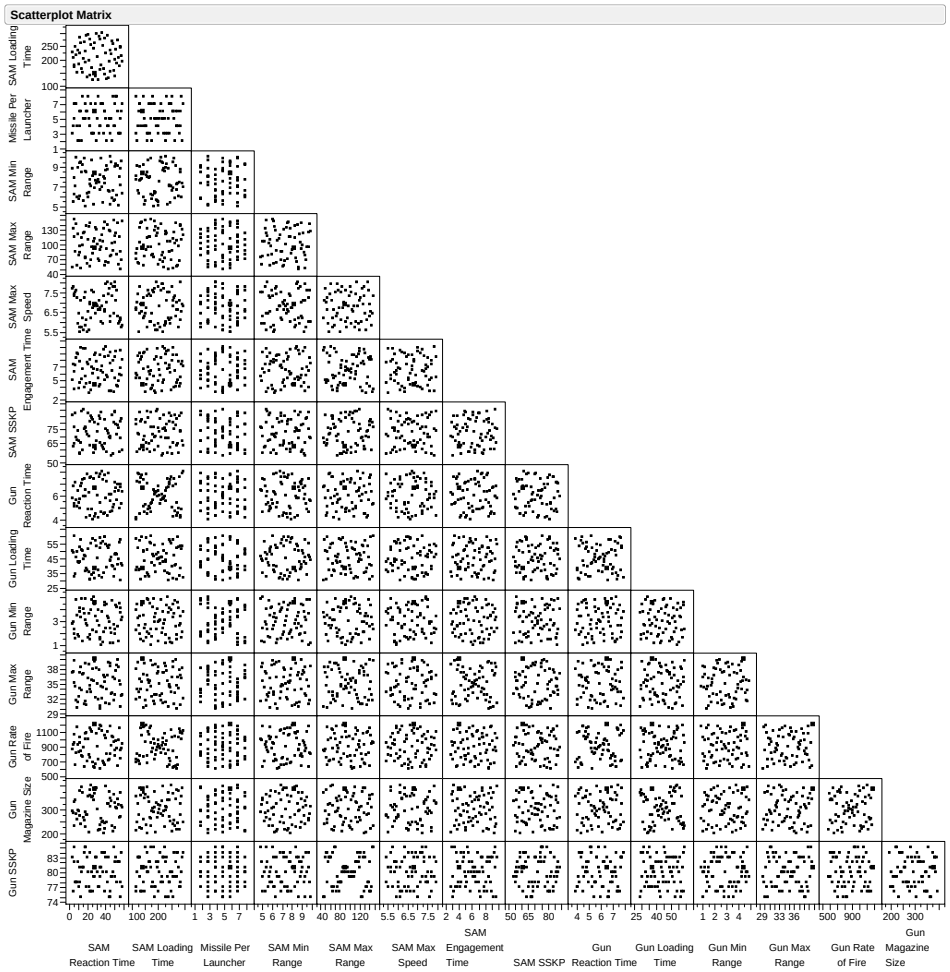


Figure 62 Scatterplot Matrix for the Experiment

2. Data Collection

The values in the design points proposed by the NOLH spreadsheet were fed into the model. For each design point, the model was run 50 times to get an unbiased mean value for the number of leakages. The results collected are then analyzed using the JMP statistical package. For each scenario, the partition model and the regression model were created and analyzed.

3. Analysis of the Defender's Scenario

The partition tree for the defender's analysis is shown in Figure 63 From the partition tree, it is observed that the two most significant factors are SAM Max Range and SAM Reaction Time. The two gun parameters, gun magazine size and gun min range, that appears at the bottom of the partition tree are not very significant as can be seen from their mean values.

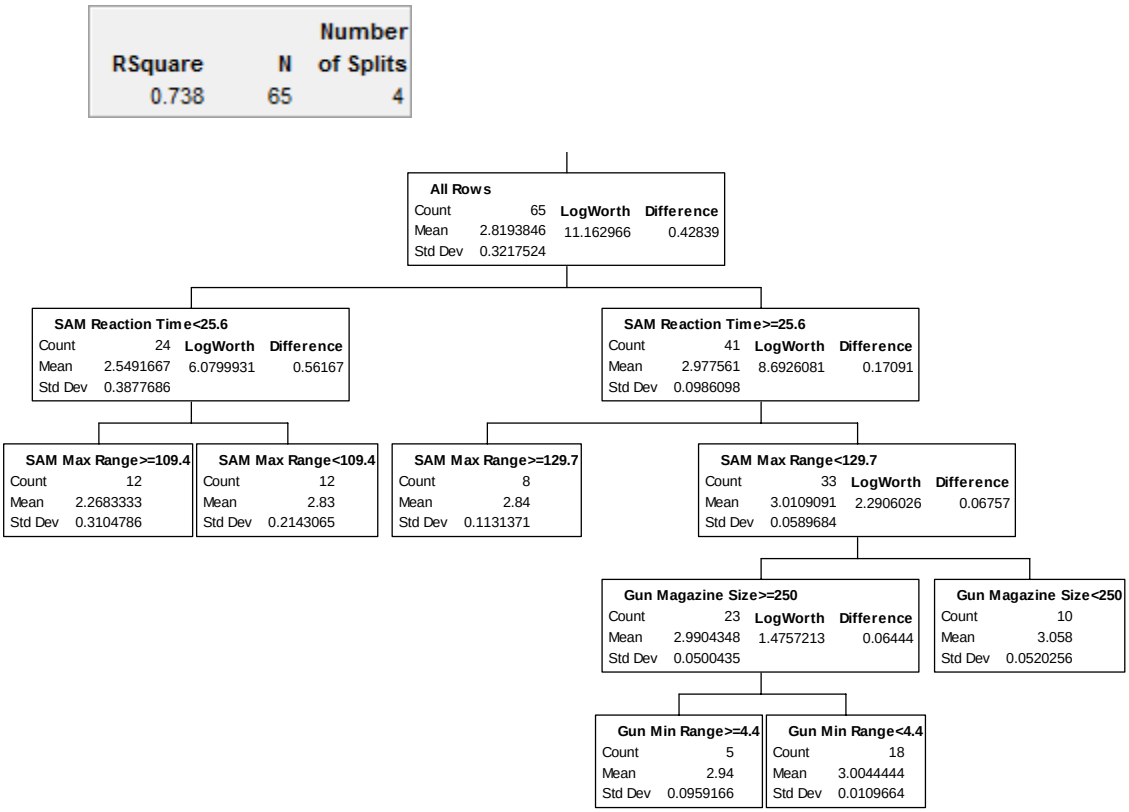


Figure 63 Partition Tree for Original Defender's Scenario

The reason there are no significant gun parameters affecting the MOE could be caused by the fact that there is uneven number of aircrafts entering the SAM ranges and the gun ranges (3 entering SAM ranges versus 2 entering gun ranges), which would directly affect the number of kills by the gun and indirectly skewed the importance of the

gun's parameter. To overcome this problem, the strike plan for the defender's scenario was revised to that shown in Figure 64, which schedules an equal number of aircrafts to fly into the anti-air gun range and the SAM range.

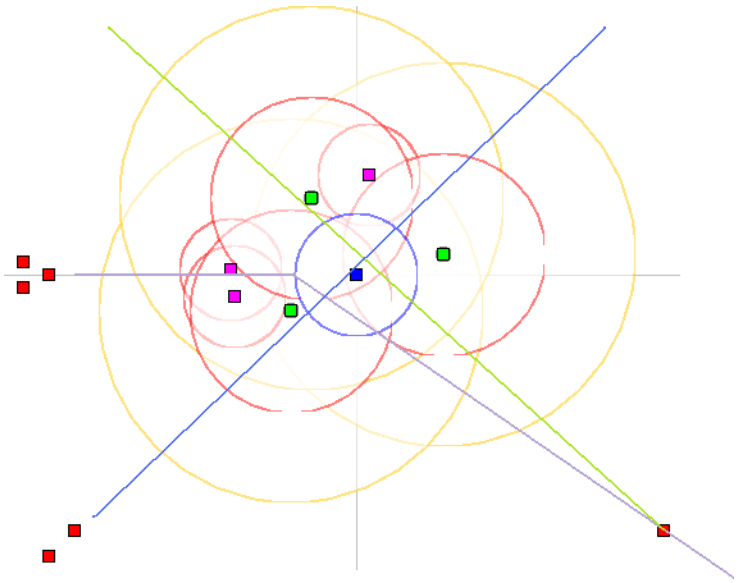


Figure 64 The Revised Defender's Scenario

The partition tree for the revised scenario is as shown in Figure 65. In the partition tree, it can be observed that on top of the two significant SAM parameters, there are also two significant gun parameters appearing in the partition tree, namely, the gun SSKP and the gun reaction time. This has confirmed the initial suspicion on the reason why there were no gun significant parameters.

On average about 4 enemy aircrafts (out of maximal 6) managed to get through the AA defenses system and reaches the bomb release line, this means on average less than 50 percent of the attackers are destroyed before reaching the target.

The best combination of weapon system's parameters for the defender is represented by the lower left branch (SAM reaction time < 32.5s, SAM max range >= 96.9 or 9.69km), which will reduce the mean number of leakages to about 2.9, with a relatively low variability (standard deviation of 0.298). Thus, in the worst case scenario, there will be at most four attacking aircrafts that are able to get through the defenses.

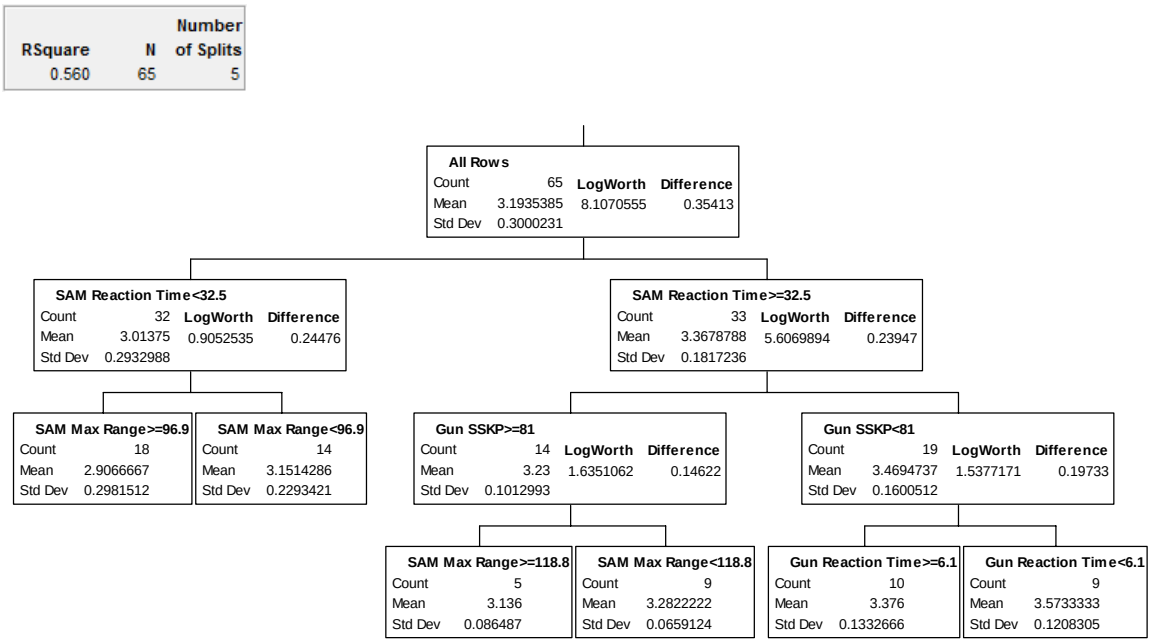


Figure 65 Partition Tree for Revised Defender's Scenario

The regression model is shown in Table 7. The significant factors are similar to those in the partition model, except for the gun magazine size.

Summary of Fit

RSquare	0.470085
RSquare Adj	0.434757
Root Mean Square Error	0.225565
Mean of Response	3.193538
Observations (or Sum Wgts)	65

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	5.8079549	0.779649	7.45	<.0001*
SAM Reaction Time	0.00898	0.001736	5.17	<.0001*
SAM Max Range	-0.00329	0.000955	-3.45	0.0010*
Gun Magazine Size	-0.001097	0.000477	-2.30	0.0250*
Gun SSKP	-0.028096	0.009426	-2.98	0.0042*

Table 7 Regression Model for Revised Defender's Scenario

Since the regression model does not perform better in terms of R square value, it might be better to use the much simpler partition model to convey the message to decision makers.

4. Analysis of the Attacker's Scenario

The data collected was fit into a partition tree as shown in Figure 66. It can be observed that the mean number of leakages is quite close to the maximal of 5 with rather small variance. The plan shows higher chances of success than the arbitrary "dumb" plan that flies the aircraft right through the anti-air gun.

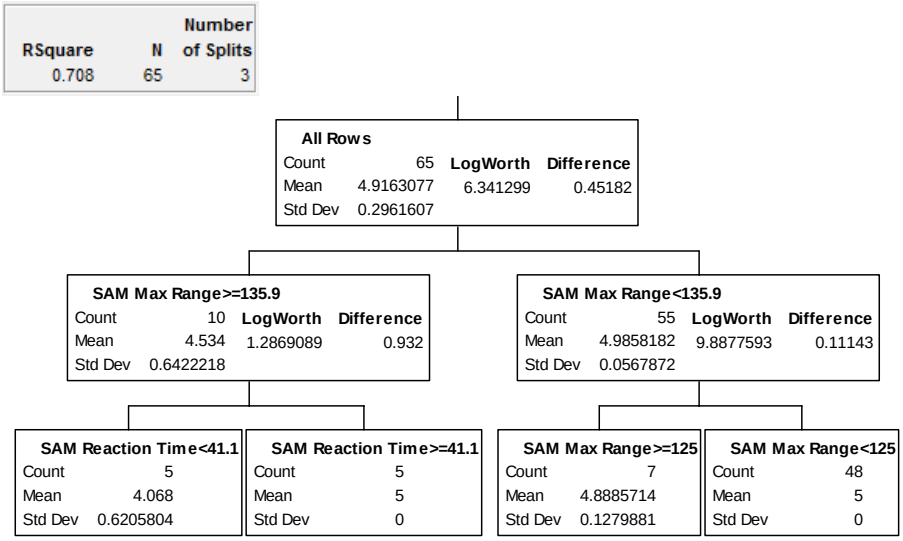


Figure 66 Partition Tree for Attacker's Scenario

There are a total of two branches that would guarantee with certainty (standard deviation of 0) that all 5 aircrafts will complete their mission. For instance, if the attacker feels his intelligence knows the SAM range and SAM reaction time accurately, and it says those are below 125 (12.5km) and above 41.1 (41.1s) respectively, then the attacker should feel very confident in the success of the mission.

A worst case scenario from the attacker's point of view, is the lower left branch with a mean of 4.0068 (on average, about one aircraft is lost). However, since the variability is not too big (standard deviation of 0.62), the attacker knows that even under this worst case scenario there is a high probability of at least three out of five aircrafts would achieve their goal.

If the attacker's only consideration is to get at least one plane to the target, the data analysis clearly shows that this objective is higher achievable, although the attacker might lose some aircrafts while doing so.

It can be observed from Figure 66 that the anti-air gun parameters have no effect on the MOE. This is as expected, since the route picked by the agent has avoided the anti-air guns.

The regression model of the data collected is shown in Table 8. The main effects showed up in the model match that of the partition model.

Summary of Fit

RSquare	0.436364
RSquare Adj	0.408644
Root Mean Square Error	0.227747
Mean of Response	4.916308
Observations (or Sum Wgts)	65

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	5.1386565	0.115436	44.52	<.0001*
SAM Max Range	-0.004058	0.000964	-4.21	<.0001*
SAM Reaction Time	0.0056442	0.001752	3.22	0.0020*
(SAM Reaction Time-32.5015)*(SAM Max Range-100.006)	0.0002371	5.421e-5	4.37	<.0001*

Table 8 Regression Model of Attacker's Scenario

5. Conclusion of the Experiment

From the analysis results of the defender point of view, it would appear that there is a high probability of at least 4 leakages into the air defense area, therefore the defender is suggested to ensure that the defended site is able to withstand the attack of least 4 strikes.

The defender should invest in purchasing SAM systems with low Reaction time (<32.5), high SAM Max Range (>96.9) and high gun SSKP (> 81).

From the attacker's point of view, the chance of having four out of five planes achieving their goal is high, although the attacker might lose a plane or two in their attack. If the attacker is very sensitive to casualties, he should reconsider his mission if he thinks there are high chances that the SAM's Range is high (> 135.9) and the SAM's Reaction time is low (< 41.1), as the defender's weapon system is the strongest with such parameters and the attacker may suffer more losses.

V. CONCLUSION AND RECOMMENDATIONS

A. CONCLUSION

The objectives of this thesis were achieved with the successful development of the low resolution air defense model using Discrete Event Simulation and the agent-based air strike plans generator. The experiment conducted has shown that agent generated plans have the ability to exploit weak spots in air defense plans, which makes it a valuable tool for foreseeing the action, reaction and counteraction dynamics between the attack and defense plans. In addition, the experiment has also shown potential ways of using both the DES engine and plans generator in answering operations research questions. It is hoped that the tools developed in this thesis could be further refined to assist air defense planners in creating consistent and highly robust defense plans.

B. RECOMMENDATIONS AND FUTURE WORK

The LEGO framework adopted in the design of the DES engine allows individual components to be further refined with little or no impact to other components in the system. The sensors used in this thesis are mainly constant time based or simple cookie-cutter based. While they have served well to facilitate the rapid construction of a proof-of-concept (POC) model for this thesis, the sensors should be refined to reflect more realistic sensor characteristics in an actual air defense setup. Potential enhancements include, modeling sensor footprint of irregular shapes and modeling sensor detection/undetection time using the glimpse model. With the framework, the sensors could be replaced with minimal effort.

To keep this POC model simple, the altitude of aircraft and terrain were not considered in the model. While modeling altitude as a continuous variable is more realistic, the introduction of a third dimension is likely to make the model much more complex. Depending on situation, it might be worthwhile to consider abstracting the altitude into discrete height intervals instead of a continuous variable to reduce the complexity of the model.

In addition to altitude, acceleration was not considered explicitly in the model. Before the model is extended, one might want to consider if acceleration is necessary for a low resolution model. It is always a good practice to keep the model simple.

For simplicity, the SAM sensors in the current model, acquire a lock on incoming aircraft based on first-come-first-served principle. The sequence of aircrafts entering the lock-on range determines the order of how the aircrafts are being locked. Although simple, this behavior might not represent air defense doctrines accurately. The model could be enhanced to assess the threat level of incoming aircraft before deciding to lock on to it or to switch its lock to another more threatening aircraft. For example, if an aircraft is in the lock-on range but not heading towards the BRL, while another aircraft is heading towards the BRL at a high velocity, the sensor might want to lock-on to the later aircraft instead of the first, even though it is in the lock-on range.

For the agent-based model, the path finding algorithm can be improved further by including the additional cost factor such as duration of the exposure to air defenses which is not currently taken into consideration. In addition, the cost of using A* algorithm can be very expensive as the area of operation for the air formation is expanded. Therefore, a dynamic area of operation should be used for each air formation; this will allow each formation to focus on its own area of operation. Hierarchical path-finding can also be use to reduce the search complexity of the path finding, this is where the entire map of the area of operation is abstracted in several levels and into linked local clusters, where, at the global level, path finding through clusters is traversed in a single big step and the search path is further refine at the cluster level of the abstracted map, which has more details, as it approach its goal.

The agent application can also take in terrain information such as DTED map or vegetation information in form of Shape files for its path finding algorithm. This will make the path planning more viable for use in modeling a real-world environment.

For the individual agent aircraft behavior model, the current implementation only caters to a few actions that the agent can do. Improvement can be made by expanding more actions to allow more dynamic agent behavior. Furthermore, sophisticated behavior

can be implemented to consider information of current position, air-defense site position or even additional incoming threats by using techniques such as a neural network to learn from past actions or Bayesian network to perform inference.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] D. Ahner, A. Buss, & J. Ruck, "Using a Low-Resolution Entity Model for Shaping Initial Conditions for High-Resolution Combat Models," in *Proceedings of the 2007 Winter Simulation Conference*, 2007, pp. 1344-1352.

[2] A. Buss, and P. Sanchez, "Simple Movement and Detection in Discrete Event Simulation," in *Proceedings of the 2005 Winter Simulation Conference*, 2005, pp. 992-1000.

[3] A. Buss, and D. Ahner, "Dynamic Allocation of Fires and Sensors (DAFS): A Low-Resolution Simulation for Rapid Modeling," in *Proceedings of the 2006 Winter Simulation Conference*, 2006, pp. 1357-1364.

[4] L. Andrew, "Hierarchical AI," http://www-cs-students.stanford.edu/~amitp/Articles/HierarchalAI.html, September 19, 2008.

[5] E. Mark, "The Clash of Civilizations," http://clash.apolyton.net/models/Model-AI.shtml, September 18, 2008.

[6] R. Straatman, W. Sterren, & A. Beij. (2006), "Killzone's AI: dynamic procedural combat tactics," http://www.cgf-ai.com/docs/straatman_remco_killzone_ai.pdf, September 18, 2008.

[7] W. Sterren, "Tactical Path-Finding with A*," in *Game Programming Gem 3*, Boston: Charles River Media, 2002, pp. 294-306.

[8] A. Buss, and P. Sanchez, "Building Complex Models with LEGOS," in *Proceedings of the 2002 Winter Simulation Conference*, 2002, pp.732-737.

[9] A. Buss, "Discrete event programming with Simkit," *Simulation News Europe*, (32/33), pp. 15-26, 2001.

[10] A. Buss, "Component-Based Simulation Modeling," *Proceedings of the 2000 Winter Simulation Conference*, 2000, pp. 964-971.

[11] M. Makoto, and N. Takuji, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 8, (1), pp. 3-30, 1998.

[12] "US Army Field Manual 44-43: Bradley Stinger Fighting Vehicle Platoon and Squad Operations," *http://www.globalsecurity.org/military/library/policy/army/fm/44-43/index.html*, November 2, 2008.

[13] D. Reece, "Movement Behavior for Soldier Agents on a Virtual Battlefield," *Massachusetts of Technology Presence*, Vol.12, (4), pp. 387-410, 2003.

[14] T. Cioppa, and T. Lucas, “Efficient Nearly Orthogonal And Space-Filling Experimental Designs For High-Dimensional Complex Models,” PhD’s Dissertation, United States Naval Postgraduate School, Monterey, CA, September 2002.

[15] S. Sanchez, “NOLHdesigns spreadsheet,” *http://diana.cs.nps.navy.mil/SeedLab/*, September 18, 2008.

APPENDIX A: LEGO MODEL OF THE DES ENGINE

This appendix shows the LEGO model of the DES engine. The details within each element are left out in the diagram as the intent of this diagram is to illustrate the relationship between the components in the model.

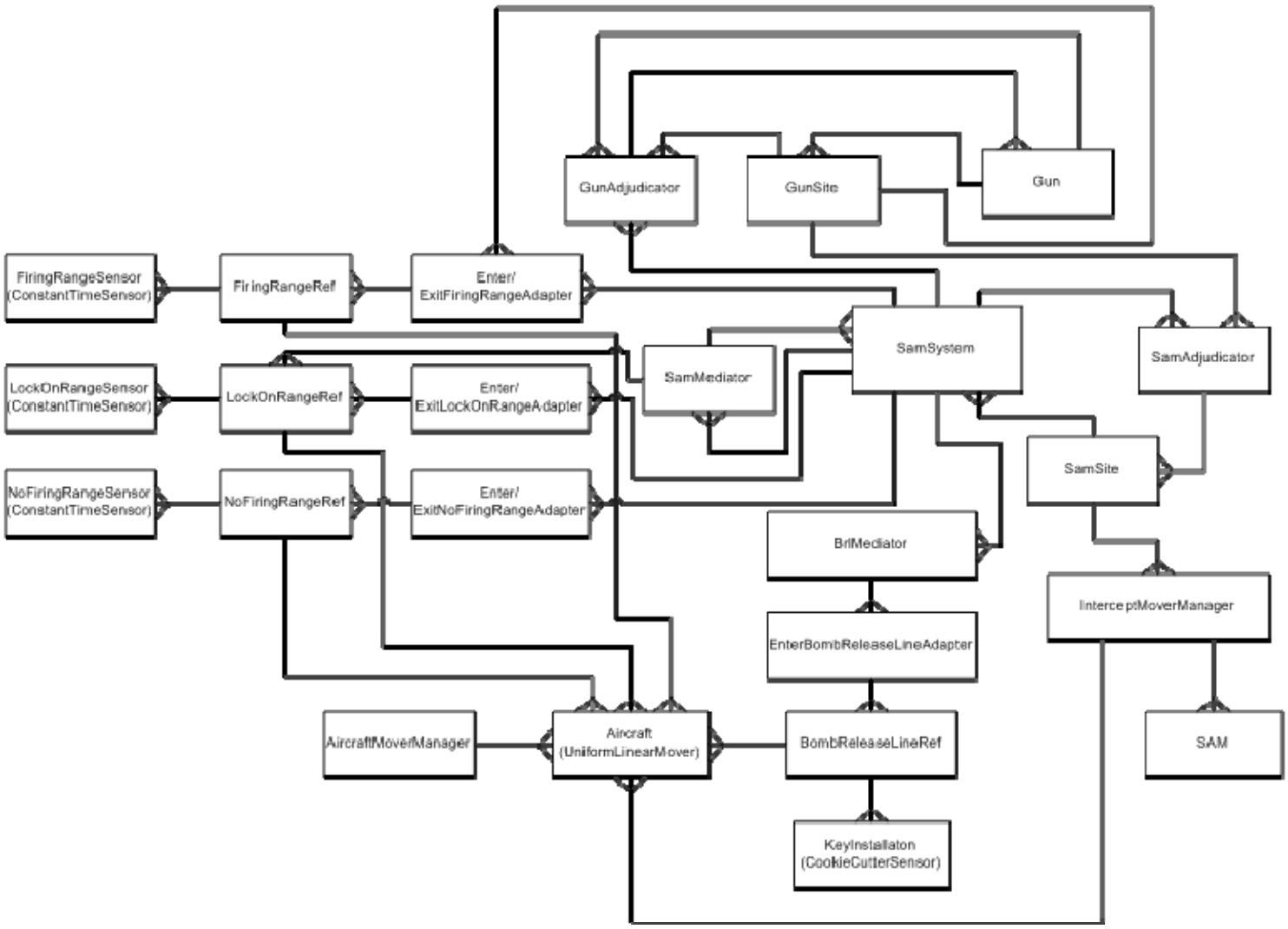


Figure 67 LEGO Model of the DES Engine

APPENDIX B: DESIGN POINTS IDENTIFIED USING NOLH DESIGN

name	SAM Reaction Time	SAM Loading Time	Missile Per Launcher	SAM Min Range	SAM Max Range	SAM Max Speed	SAM Engagement Time	SAM SSKP	Gun Reaction Time	Gun Loading Time	Gun Min Range	Gun Max Range	Gun Rate of Fire	Gun Magazine Size	Gu
1	48.5	128.4	4	6.6	62.5	7.4	6.6	72	7.9	51.6	3.2	39.4	713	325	
2	57.4	249.4	3	7.1	84.4	6.1	6.5	61	6.9	57.7	4.1	34.6	741	259	
3	54	184.7	8	6.1	79.7	7.6	4.1	71	5.6	47.3	4.2	35.8	881	303	
4	40.2	280.3	6	7.3	95.3	6.6	4.9	64	4.5	59.5	4.6	36.3	647	241	
5	55.7	204.4	3	5.1	59.4	5.9	4.8	76	6.4	44.5	1.3	38.4	984	319	
6	34.2	285.9	3	7.4	65.6	7.1	4	65	7.6	33.3	2.6	37.8	1134	213	
7	47.1	153.8	5	5.2	75	6.3	9.2	70	4.6	43.1	2.4	39.1	1172	347	
8	49.7	257.8	6	6.6	87.5	7.6	7.9	56	5	39.4	1.1	36.4	1031	206	
9	42.8	126.5	4	9.1	90.6	7	5.7	58	7.4	38.9	4.4	33.9	797	247	
10	58.3	243.9	5	8.8	50	6.2	9.5	68	6.7	30	3.7	30.3	853	369	
11	33.4	122.3	8	7.7	70.3	6.6	5.2	79	4.1	37	3.9	31.9	731	209	
12	59.1	212.8	6	9.6	67.2	5.7	5.6	63	5.3	41.3	3.1	32.5	672	316	
13	36.1	159.4	4	8.5	89.1	6.7	4.3	68	7.8	58.6	1.6	30.5	1191	234	
14	48	218.4	5	9.5	76.6	7.9	3.3	58	6.1	46.9	2	33	909	334	
15	36.8	179.1	7	9.8	93.8	6	6.3	68	4.6	54.8	1	34.4	1181	289	
16	42	224.1	5	8.6	64.1	7.9	10	75	5.8	54.4	2.4	31.1	1013	397	
17	52.3	196.8	4	5.6	117.2	8	8.9	89	6.3	47.8	2.3	32.8	844	328	
18	38.9	274.7	4	6.7	107.8	6.6	9.8	83	5.4	32	1.4	30	600	235	
19	48.4	190.3	3	9.4	129.1	7.7	7.4	69	6.1	59.9	2.9	39.8	744	379	
20	38.5	252.2	7	5.3	104.7	5.9	6.1	61	7.3	56.3	1.2	34.2	703	250	
21	50.5	156.6	4	5.5	148.4	6.4	3.1	80	5.2	52.8	2.7	31.7	966	363	
22	45.8	269.1	2	5.8	101.6	7.3	6.4	86	4.7	40.3	4.7	33.1	1041	228	
23	60	193.1	7	6.3	146.9	6.8	8	59	6.3	39.8	3.3	32.7	938	309	
24	37.7	300	7	6.9	116.6	6.8	7	63	8	31.9	3.8	32	1116	244	
25	39.4	165	3	8	131.3	7.3	9.6	60	4.4	34.7	2.1	36.6	675	200	
26	46.3	232.5	3	10	126.6	6	7.5	68	5.5	43.6	1.1	38.6	806	313	
27	51.4	136.9	7	9	103.1	7.8	5.3	88	7.1	37.5	1.9	35.9	778	222	
28	54.6	271.9	6	9.1	142.2	6.3	3.9	78	6.9	40.8	2.5	39.8	638	344	
29	56.6	170.4	3	8	114.1	6	8.8	66	4.9	55.8	4.6	35.8	1144	283	
30	43.7	288.8	4	8.7	139.1	8	5.4	74	4.3	53.4	3.4	36.7	1106	322	
31	41.1	142.5	6	7.5	145.3	6.5	6.4	82	7.7	59.1	4.3	35.3	1050	219	
32	58.1	236.1	8	9.3	121.9	7.1	9.3	79	6.6	48.3	4.3	37.7	975	366	
33	32.5	210	5	7.5	100	6.8	6.5	73	6	45	3	35	900	300	
34	50.5	261.6	6	8.4	137.5	6.1	4.4	73	4.1	38.4	2.8	30.6	1088	275	
35	7.6	170.6	7	7.9	115.6	7.4	6.2	64	5.1	32.3	1.9	35.2	1089	331	
36	11	238.3	2	8.9	126.3	8.9	8.9	74	6.4	42.7	1.8	31.3	919	297	
37	24.6	139.7	4	7.7	143.8	6.9	6.1	61	7.5	50.5	1.4	33.6	1153	359	
38	9.3	215.6	7	9.9	140.6	7.6	8.3	69	5.6	45.5	4.8	31.6	816	291	
39	30.8	134.1	7	7.6	134.4	6.4	9	60	4.6	56.7	3.4	32.2	666	366	
40	17.9	266.3	5	9.8	125	7.2	3.8	75	7.4	46.9	3.6	30.9	628	253	
41	18.3	165.2	2	8.4	112.5	6.7	6.1	89	7	60.6	4.9	33.6	769	394	
42	22.2	294.4	6	5.9	109.4	6.6	7.3	67	4.8	51.1	1.6	36.1	1003	353	
43	57	176.3	5	6.2	150	7.3	3.5	77	8.3	60	2.3	39.7	947	231	
44	31.6	297.2	2	7.3	129.7	6.9	7.8	66	7.9	53	2.1	38.1	1069	391	
45	5.9	207.2	4	5.4	132.8	7.8	7.4	62	6.8	48.8	2.9	37.5	928	284	
46	29.9	260.6	6	6.8	110.9	6.8	8.7	80	4.2	51.4	4.4	35.5	809	366	
47	17	201.6	5	5.5	123.4	5.6	9.7	90	5.9	44.1	4	37	891	266	
48	26.2	240.9	3	5.2	106.3	7.5	6.7	57	7.4	55.2	5	35.6	619	341	
49	23	195.9	5	5.4	135.9	5.5	3	70	6.2	35.6	3.6	35.9	755	203	
50	12.7	221.3	6	9.4	82.8	6.5	4.5	56	8.7	49.2	3.8	37.2	956	272	
51	29.1	145.3	6	8.3	92.2	6.9	4.2	62	6.6	38	4.6	40	1200	384	
52	19.6	229.7	5	8.5	71.9	5.8	9.8	76	5.9	36.1	3.1	39.2	1078	228	
53	26.5	167.8	3	9.7	95.3	7.6	6.9	84	4.6	33.6	4.8	35.6	1097	350	
54	14.5	263.4	6	9.5	51.6	7.1	9.9	65	6.9	57.2	3.3	35.3	834	238	
55	16.2	190.9	6	9.2	98.4	6.2	6.6	59	7.3	49.7	1.3	36.9	759	372	
56	5	226.9	3	8.8	53.1	7.7	5	86	8.8	55.2	2.8	37.3	863	291	
57	37.3	120	3	8.1	81.3	6.7	6	82	4	58.1	2.2	38	684	366	
58	28.6	289	7	7	68.8	6.2	3.4	65	7.6	58.3	3.9	33.4	1128	400	
59	18.8	187.5	7	8	73.4	7.5	8.8	77	6.5	46.4	4.9	31.4	994	288	
60	13.6	283.1	4	6	96.9	5.7	7.7	57	4.9	52.5	4.1	34.1	1022	375	
61	10.2	146.1	4	5.9	87.8	7.2	9.1	67	5.1	49.2	3.5	30.2	1163	266	
62	8.4	246.8	7	7	85.9	7.5	7.2	87	7.1	34.2	1.8	34.5	699	339	
63	21.3	131.3	6	6.3	60.9	5.8	7.8	71	7.8	36.6	2.8	33.3	894	278	
64	23.9	277.8	4	7.2	94.7	7	4.8	63	4.3	30.9	1.9	34.7	790	381	
65	11.9	181.9	2	8.7	76.1	6.4	3.7	66	5.4	41.7	1.7	32.3	825	294	

Table 9 Design Points Identified using NOLH Design

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C: SAMPLE STRIKE PLAN

A sample of the agent-generated strike plan comprises of five strike aircrafts is shown below.

```
<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE AIR_STRIKE_PLANNING []>
<AIR_STRIKE_PLANNING xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="AD Strike Schema.xsd">
        <!-- DES Engine Air Strike Plan -->
        <!-- Naval Postgraduate School -->
        <VERSION_NO>1.0.0</VERSION_NO>
        <FILE_DATE>21:09:28 PDT</FILE_DATE>
        <FILE_TIME>21:09:28</FILE_TIME>
        <AIR_STRIKE_PLAN>
                <AIRCRAFT_TASKS>
                        <AIRCRAFT_TASK>
                                <REQUIRED_UNIT>
                                        <CALLSIGN>EAGLE1</CALLSIGN>
                                        <DESCRIPTION>F16</DESCRIPTION>
                                        <UNIT_ID>AS320</UNIT_ID>
                                        <UNIT/>
                                        <UNIT_ROLE>AIR_STRIKE</UNIT_ROLE>
                                        <MAX_SPEED>3</MAX_SPEED>
                                </REQUIRED_UNIT>
                                <WAYPOINTS>
                                        <WP1>-220 -200 2</WP1>
                                        <WP2>-75 -75 2.5</WP2>
                                        <WP3>-30 -30 3</WP3>
                                        <WP4>-100 -20 2.5</WP4>
                                        <WP5>220 220 2</WP5>
                                </WAYPOINTS>
                        </AIRCRAFT_TASK>
                        <AIRCRAFT_TASK>
                                <REQUIRED_UNIT>
                                        <CALLSIGN>EAGLE2</CALLSIGN>
                                        <DESCRIPTION>F16</DESCRIPTION>
                                        <UNIT_ID>AS310</UNIT_ID>
                                        <UNIT/>
                                        <UNIT_ROLE>AIR_STRIKE</UNIT_ROLE>
                                        <MAX_SPEED>3</MAX_SPEED>
                                </REQUIRED_UNIT>
                                <WAYPOINTS>
                                        <WP1>-240 -220 2</WP1>
                                        <WP2>-160 -140 3</WP2>
                                        <WP3>-20 0 3</WP3>
                                        <WP4>-220 220 2</WP4>
                                </WAYPOINTS>
                        </AIRCRAFT_TASK>
                        <AIRCRAFT_TASK>
```

```
                    <REQUIRED_UNIT>
                        <CALLSIGN>EAGLE3</CALLSIGN>
                        <DESCRIPTION>F16</DESCRIPTION>
                        <UNIT_ID>AS300</UNIT_ID>
                        <UNIT/>
                        <UNIT_ROLE>AIR_STRIKE</UNIT_ROLE>
                        <MAX_SPEED>3</MAX_SPEED>
                    </REQUIRED_UNIT>
                    <WAYPOINTS>
                        <WP1>-240 0 2.5</WP1>
                        <WP2>-100 0 3</WP2>
                        <WP3>240 -140 2.5</WP3>
                    </WAYPOINTS>
                </AIRCRAFT_TASK>
                <AIRCRAFT_TASK>
                    <REQUIRED_UNIT>
                        <CALLSIGN>EAGLE4</CALLSIGN>
                        <DESCRIPTION>F16</DESCRIPTION>
                        <UNIT_ID>A301</UNIT_ID>
                        <UNIT/>
                        <UNIT_ROLE>AIR_STRIKE</UNIT_ROLE>
                        <MAX_SPEED>3</MAX_SPEED>
                    </REQUIRED_UNIT>
                    <WAYPOINTS>
                        <WP1>240 -200 2.5</WP1>
                        <WP2>0 0 3</WP2>
                        <WP3>240 220 2</WP3>
                    </WAYPOINTS>
                </AIRCRAFT_TASK>
                <AIRCRAFT_TASK>
                    <REQUIRED_UNIT>
                        <CALLSIGN>EAGLE5</CALLSIGN>
                        <DESCRIPTION>F16</DESCRIPTION>
                        <UNIT_ID>A311</UNIT_ID>
                        <UNIT/>
                        <UNIT_ROLE>AIR_STRIKE</UNIT_ROLE>
                        <MAX_SPEED>3</MAX_SPEED>
                    </REQUIRED_UNIT>
                    <WAYPOINTS>
                        <WP1>-260 -10 2</WP1>
                        <WP2>-100 -10 2.5</WP2>
                        <WP3>240 -150 2</WP3>
                    </WAYPOINTS>
                </AIRCRAFT_TASK>
            </AIRCRAFT_TASKS>
    </AIR_STRIKE_PLAN>

    </AIR_STRIKE_PLANNING>
```

APPENDIX D: SAMPLE AIR DEFENSE PLAN

A sample air defense plan consisting of three anti-air guns and three SAMs is shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<AIR_DEFENSE_PLANNING xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="AD
Plan Schema.xsd">
        <!-- DES Engine Air Defense Plan -->
        <!-- Naval Postgraduate School -->
        <VERSION_NO>1.0.0</VERSION_NO>
        <FILE_DATE>21:09:28 PDT</FILE_DATE>
        <FILE_TIME>21:09:28</FILE_TIME>
        <ADA_SYSTEM>
                <ADA_PLANS>
                        <ADA_PLAN>
                                <PLAN_NAME>AD Plan</PLAN_NAME>
                                <DESCRIPTION>Air Defense Plan</DESCRIPTION>
                                <DATETIME>39665.88066207176</DATETIME>
                                <ADA_TASKS>
                                        <ADA_TASK>
                                                <REQUIRED_UNIT>
                                                        <CALLSIGN/>
                                                        <DESCRIPTION/>
                                                        <UNIT_ID>ADA 1</UNIT_ID>
                                                        <UNIT/>
                                                        <UNIT_TYPE>GUN I</UNIT_TYPE>
                                                </REQUIRED_UNIT>
                                                <LATITUDE>-17.5</LATITUDE>
                                                <LONGITUDE>-95</LONGITUDE>
                                                <LOCATION_NAME>LOCATION 1</LOCATION_NAME>

        <MAIN_AREA_OF_RESPONSIBILITY>3</MAIN_AREA_OF_RESPONSIBILITY>

        <ALT_AREA_OF_RESPONSIBILITY_1>2</ALT_AREA_OF_RESPONSIBILITY_1>

        <ALT_AREA_OF_RESPONSIBILITY_2>4</ALT_AREA_OF_RESPONSIBILITY_2>
                                                <READY_TIME>39665.88066243056</READY_TIME>
                                        </ADA_TASK>
                                        <ADA_TASK>
                                                <REQUIRED_UNIT>
                                                        <CALLSIGN/>
                                                        <DESCRIPTION/>
                                                        <UNIT_ID>ADA 2</UNIT_ID>
                                                        <UNIT/>
                                                        <UNIT_TYPE>GUN I</UNIT_TYPE>
                                                </REQUIRED_UNIT>
                                                <LATITUDE>4</LATITUDE>
                                                <LONGITUDE>-97.5</LONGITUDE>
                                                <LOCATION_NAME>LOCATION 2</LOCATION_NAME>

        <MAIN_AREA_OF_RESPONSIBILITY>4</MAIN_AREA_OF_RESPONSIBILITY>

        <ALT_AREA_OF_RESPONSIBILITY_1>1</ALT_AREA_OF_RESPONSIBILITY_1>

        <ALT_AREA_OF_RESPONSIBILITY_2>3</ALT_AREA_OF_RESPONSIBILITY_2>
                                                <READY_TIME>39665.88066243056</READY_TIME>
                                        </ADA_TASK>
                                        <ADA_TASK>
                                                <REQUIRED_UNIT>
                                                        <CALLSIGN/>
                                                        <DESCRIPTION/>
                                                        <UNIT_ID>ADA 3</UNIT_ID>
```

```
                                        <UNIT/>
                                        <UNIT_TYPE>GUN I</UNIT_TYPE>
                                    </REQUIRED_UNIT>
                                    <LATITUDE>77.6</LATITUDE>
                                    <LONGITUDE>10.3</LONGITUDE>
                                    <LOCATION_NAME>LOCATION 3</LOCATION_NAME>
                                    <READY_TIME>39665.88066243056</READY_TIME>
                                </ADA_TASK>
                                <ADA_TASK>
                                    <REQUIRED_UNIT>
                                        <CALLSIGN/>
                                        <DESCRIPTION/>
                                        <UNIT_ID>ADA 4</UNIT_ID>
                                        <UNIT/>
                                        <UNIT_TYPE>SAM TYPE I</UNIT_TYPE>
                                    </REQUIRED_UNIT>
                                    <LATITUDE>60</LATITUDE>
                                    <LONGITUDE>-35</LONGITUDE>
                                    <LOCATION_NAME>LOCATION 4</LOCATION_NAME>

<MAIN_AREA_OF_RESPONSIBILITY>1</MAIN_AREA_OF_RESPONSIBILITY>

<ALT_AREA_OF_RESPONSIBILITY_1>2</ALT_AREA_OF_RESPONSIBILITY_1>

<ALT_AREA_OF_RESPONSIBILITY_2>4</ALT_AREA_OF_RESPONSIBILITY_2>

                                    <READY_TIME>39665.88066243056</READY_TIME>
                                </ADA_TASK>
                                <ADA_TASK>
                                    <REQUIRED_UNIT>
                                        <CALLSIGN/>
                                        <DESCRIPTION/>
                                        <UNIT_ID>ADA 5</UNIT_ID>
                                        <UNIT/>
                                        <UNIT_TYPE>SAM TYPE I</UNIT_TYPE>
                                    </REQUIRED_UNIT>
                                    <LATITUDE>-28</LATITUDE>
                                    <LONGITUDE>-51</LONGITUDE>
                                    <LOCATION_NAME>LOCATION 5</LOCATION_NAME>
                                    <READY_TIME>39665.88066243056</READY_TIME>
                                </ADA_TASK>
                                <ADA_TASK>
                                    <REQUIRED_UNIT>
                                        <CALLSIGN/>
                                        <DESCRIPTION/>
                                        <UNIT_ID>ADA 6</UNIT_ID>
                                        <UNIT/>
                                        <UNIT_TYPE>SAM TYPE I</UNIT_TYPE>
                                    </REQUIRED_UNIT>
                                    <LATITUDE>15.3</LATITUDE>
                                    <LONGITUDE>68.3</LONGITUDE>
                                    <LOCATION_NAME>LOCATION 6</LOCATION_NAME>
                                    <READY_TIME>39665.88066243056</READY_TIME>
                                </ADA_TASK>
                            </ADA_TASKS>
                        </ADA_PLAN>
                    </ADA_PLANS>
</ADA_SYSTEM>

</AIR_DEFENSE_PLANNING>
```

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Christian J. Darken
   Naval Postgraduate School
   Monterey, California

4. Arnold H. Buss
   Naval Postgraduate School
   Monterey, California

5. Professor Yeo Tat Soon, Director
   Temasek Defence Systems Institute
   National University of Singapore
   Singapore

6. Tan Lai Poh (Ms), Assistant Manager
   Temasek Defence Systems Institute
   National University of Singapore
   Singapore